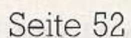

MAI 1985
OS 55,—/Sfr. 6,50 Lit 5.500/hfl 8,—/dkr 30,—
DM 6,50
64'er
5|85
DAS MAGAZIN FÜR COMPUTER-FANS
Eine Markt & Technik Publikation
Alle Matrixdrucker für den C 64
★ Vergleichstest ★ Auswahlhilfe
★ Marktübersicht
Alles über Dateiverwaltung
Moderne Programmiersprachen
★ Grundlagen
★ Test: Ada und Logo für C 64
C 64 an der Stereoanlage
Bauanleitung:
Mit 5 Mark sind Sie dabei
Programmiertips für Anfänger
DM 2.000
DM 1.400
DM 700
Steuern mit dem User-Port ★ Lösungen zu Abenteuerspielen ★ Textprogramm für 80-Zeichen-Karte ★ »Intelligenz« zum Abtippen: Das Programm, das mich versteht ★ Tips und Tricks für C 64, C 16 und VC 20

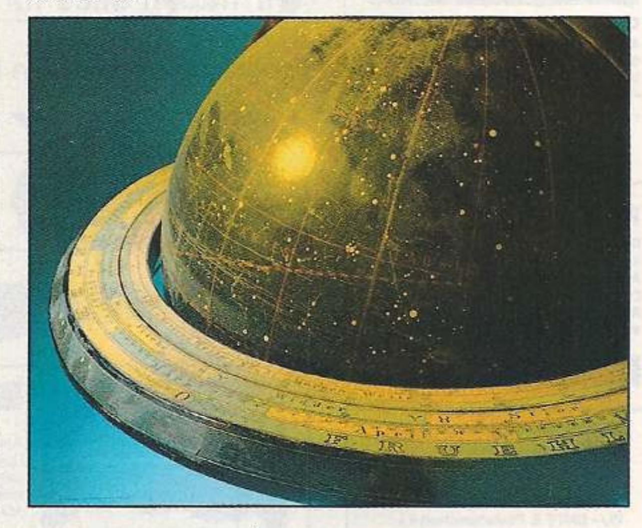

64ER ONLINE



64ER ONLINE



Seite 52



Weißt Du wieviel Sternlein stehen ...

Mit der Anwendung des Monats können Sie sich jeden Abend eine Sternenkarte zeichnen lassen, um Sternbilder, Sterne und Planeten am Nachthimmel leichter zu finden. Interessant dabei: Sie können von jedem Punkt der Erde den Sternenhimmel betrachten.
Seite 52

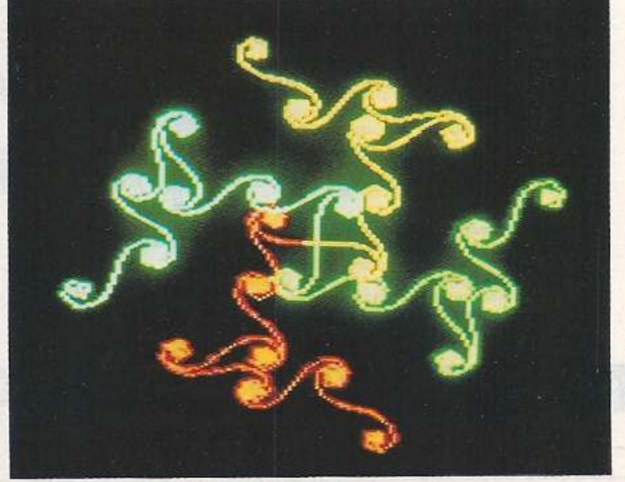
Seite 49



Infocom-Geheimnisse gelüftet?

Die Infocom-Adventures gehören zu den besten Abenteuerspielen. Um sie zu lösen, könnte man sich der angebotenen »Hint books« bedienen. Um Ihnen bei den schwierigsten Stellen zu helfen, ohne den eigentlichen Spaß zu verderben, haben wir Tips für 14 Infocom-Adventures zusammengestellt.
Seite 49

Seite 135



Logo im Test

Bunte Grafiken sind zwar gewissermaßen das Markenzeichen von Logo, aber durchaus nicht das einzige, was diese moderne Programmiersprache zu bieten hat. Listenverarbeitung und Rekursionen sind weitere Stichworte; in der C 64-Version gibt's als Zugabe noch acht Kobolde und jede Menge Spaß am Programmieren.
Seite 135

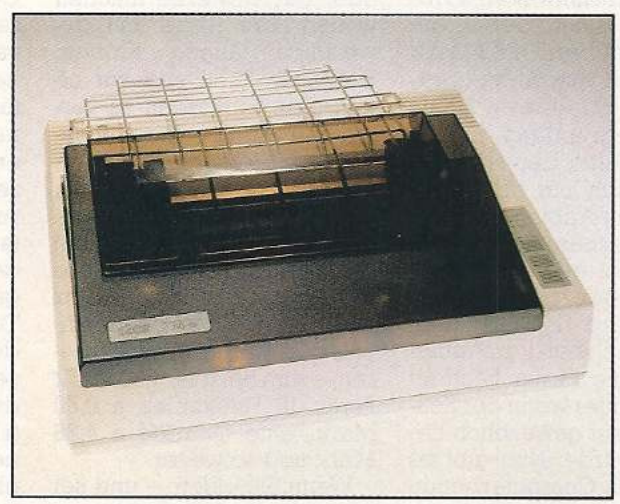
Seite 32



Die Videowerkstatt

Mit dem Digitizer VD 64 lassen sich sehr leicht Videobilder vom Videorecorder oder von der Videokamera in den Speicher des C 64 übertragen. Es stehen vier Graustufen oder Farben zur Verfügung. Beachtenswert ist die Geschwindigkeit: zwei Bilder pro Sekunde. Zum Digitalisieren kann jedes Videosignal hergenommen werden.
Seite 32

Seite 15



Alle Matrixdrucker für den C 64

Der Drucker ist neben den Massenspeichern das wichtigste Peripheriegerät für den Computer. Der meistverwendete Typ ist dabei der Matrixdrucker. Ausführliche Tests der neuesten Drucker, eine Marktübersicht und Vergleichstest der Drucker unter 700 Mark sollen Ihnen bei der Entscheidung helfen.
Seite 15

Seite 34



Mit 5 Mark zu neuen Dimensionen

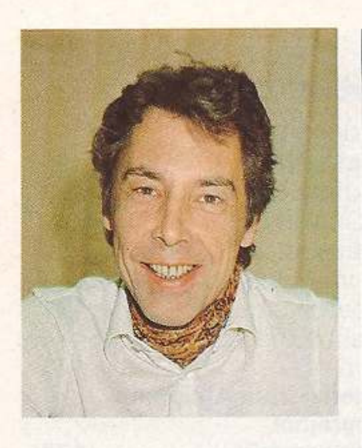
Daß eine gute Stereoanlage mehr Klangvolumen als der Lautsprecher eines Fernsehers oder Monitors hat, ist eine Binsenweisheit. Wir zeigen Ihnen daher, wie Sie mit sehr einfachen Mitteln einen C 64 an eine Stereoanlage anschließen können. Nicht nur Ghostbusters wird dadurch zum Erlebnis.
Seite 34

Wettbewerbe

Programme zum Abtippen

Kurse

Rubriken



Künstlich intelligent?

»Eliza«, die wir als Anregung für unseren Programmierwettbewerb (das Ergebnis finden Sie in dieser Ausgabe) genommen haben, ist einer der klassischen Versuche auf dem Gebiet der Künstlichen Intelligenz. Das Programm und vor allem die Reaktion des Publikums zeigten schon vor Jahren, daß der Computer für intelligent gehalten wird, weil der Mensch in die programmierten Fragen und Antworten Intelligenz, das heißt einen bestimmten Sinn hineininterpretiert. Derartige Effekte nutzte man aber schon in der Antike beim Orakel von Delphi ganz ohne Rechner...

Ein wesentliches Problem bei den Forschungen auf dem Gebiet der Künstlichen Intelligenz ist die Interpretation der natürlichen Sprache. Unsere normale Ausdrucksweise ist für einen Computer viel zu redundant (wir machen zu viele Worte) und viel zu ungenau beziehungsweise unvollständig (wir setzen zu viel als selbstverständlich bekannt voraus).

Ob es um automatische Übersetzung, akustische Spracheingabe oder Programmieren und Datenbankabfrage in natürlicher Sprache geht: Die Probleme häufen sich. Bis ein Computer richtig Deutsch versteht (Englisch kann er auch nicht besser — GET hin, PRINT her), wird noch einige Zeit vergehen. Vorläufig ist man keineswegs sicher, ob ein Computer den Bibelspruch »Der Geist ist willig, aber das Fleisch ist schwach« mit »Es fehlt nicht an guten Absichten, sondern an der Realisierung« oder mit »Der Whisky war in Ordnung, aber das Steak ließ zu wünschen übrig« übersetzen würde. Woraus Sie ersehen, daß Künstliche Intelligenz nicht nur ein Forschungsprojekt mit Tücken, sondern auch ein durchaus amüsantes Denksportthema ist.

Michael Pauly, Chefredakteur

Die neue Abmahnmasche: Vorsicht bei Programmangeboten

Die neueste Abmahn-Masche, mit der unterbeschäftigte Rechtsanwälte hart am Rande der Legalität zu Geld zu kommen suchen, trifft die Programmierer

So bekam kürzlich ein Leser, der eine selbstgeschriebene Grafik-Routine für 30 Mark in einer Kleinanzeige angeboten hatte, von einem Rechtsanwalt eine Abmahnung samt Gebührenforderung über 501,60 Mark (willkürlich vom Anwalt festgesetzter »Streitwert«: 20000 Mark). Begründung: In der Anzeige fehle der Hinweis, daß es sich um einen gewerblichen Anbieter handle — das verstoße aber gegen das Gesetz gegen unlauteren Wettbewerb.

Das wäre in Ordnung, wenn es sich bei dem Anbieter um eine Firma handeln würde — oder wenn der Softwareverkauf gewerblich betrieben würde. Nun gibt es aber viele Computerbenutzer, die zwar bereit (und vielleicht sogar interessiert sind), das eine oder andere selbstgeschriebene Programm an Interessenten abzugeben — die aber daraus keineswegs ein Geschäft oder gar Gewerbe machen wollen. Um Ärger mit gewerblich tätigen Firmen, Rechtsanwälten und vor allem dem Finanzamt zu vermeiden, sollten Sie entweder nur tauschen (Tausch zwischen Privatleuten im Rahmen ihres Hobbys ist keine gewerbliche Tätigkeit) oder darauf achten, daß Sie lediglich einen Kostenersatz berechnen. Es ist zweckmäßig, den Betrag zu spezifizieren — zum Beispiel 1,30 Mark Porto, 10 Fotokopien á 0,50 Mark, eine Diskette á 4,85 Mark und so weiter.

Wenn Sie einen — und sei er auch nur bescheiden — Gewinn erzielen wollen, müssen Sie auf schriftlichen Unterlagen in Inseraten und so weiter durch eine geeignete Angabe wie »Firma«, »Programmierbüro«, »Softwarevertrieb« oder ähnliches erkennen lassen, daß Sie sich gewerblich betätigen. Sie müssen außerdem das Gewerbe bei der Gemeinde beziehungsweise Stadt anmelden und ein Minimum an Buchführung machen, damit Sie dem Finanzamt jederzeit Umsätze, Kosten und Gewinn nachweisen können. In den meisten Fällen werden Umsatz und Ertrag so gering sein, daß ohnehin keine ernstzunehmende Menge Steuern zu bezahlen ist.

Sollten Sie als Privatmann eine Abmahnung der oben erwähnten Art bekommen, dann schreiben Sie umgehend zurück, daß Sie ihren Computer nur privat benutzen, die Programme für private Zwecke geschrieben haben und durch das Anbieten ihrer selbstgeschriebenen Progamme Kontakt zu anderen Computerbenutzern zum Zweck des Erfahrungs- und Programmaustausches suchen. Ihre selbstgeschriebenen Programme gäben Sie entweder im Tausch oder gegen Ersatz der durch Erstellen und Versenden der Kopie entstehenden Kosten ab. Falls das zutrifft, brauchen Sie auch keine Unterwerfungserklärung abzugeben und keine Gebühren zu zahlen.

(py)

Abmahnschwindler nie gefaßt

Mit einem Abmahnschwindel besonderer Art tat sich im vergangenen Jahr eine »R + S Computerorganisation« in Berlin hervor: Sie trat als angeblicher Wettbewerber auf, verlangte von zahlreichen Anbietern von Raubkopien die Abgabe einer Unterlassungserklärung sowie die Bezahlung einer Gebührenrechnung in Höhe von mehreren hundert Mark. Der Rechnungsbetrag sollte in bar zusammen mit der Unterlassungserklärung an eine Postfachadresse in Berlin geschickt werden. Die Staatsanwaltschaft schaltete sich sehr schnell ein und stellte fest, daß das angegebene Postfach in der vorgegaukelten Form nicht existierte. Es handelte sich dabei, wie die Justizpressestelle jetzt auf Anfrage mitteilte, um die Nummer einer Postlagerkarte bei einem Berliner Postamt, die tatsächlich ausgegeben worden war »ohne daß die Personalien des Empfängers notiert worden wären oder hätten notiert werden müssen«. Bei Beobachtungen in dem Postamt stellte die Kriminalpolizei im vergangenen Jahr zwar einen 15jährigen Jungen, der mit der Postlagerkarte und einer Vollmacht der Schwindelfirma die Post abholen wollte. Als die Polizisten ihn nach dem Auftraggeber fragten, deutete er auf einen etwa hundert Meter vom Postamt entfernt stehenden Mann, der daraufhin zusammen mit einem anderen, mit einem Auto die Flucht ergriff. Da der Junge nach Feststellungen der Polizei als Mittäter ausscheidet und die beiden Flüchtigen nicht identifiziert werden konnten, wurde das Verfahren gegen R + S wohl oder übel eingestellt.

(py)

Software fast umsonst

Der »Folklife Terminal Club« in den USA hat seine Software-Palette PET, CBM, C 64, VC 20, C 16 und Plus/4 auf über 6000 Programme vergrößert und auch für Nichtmitglieder zugänglich gemacht. Es handelt sich dabei um »Public Domain Software«, die von den Autoren umsonst zur Verfügung gestellt wird. Für die Diskette, den Kopieraufwand und die Versandkosten werden allerdings 15 Dollar berechnet. Als erstes sollte man sich die »Catalog Disk« für seinen Computertyp bestellen. Sie enthält alle Programme, die derzeit angeboten werden. Die Bezahlung sollte über einen Bankscheck, den auch US-Banken akzeptieren oder über Auslandspostanweisungen erfolgen.

Info: Folklife Terminal Club, Box 555-SB, Co-op City Station, Bronx, N.Y. 10475 USA

Vorsicht »B-Platine«

Für das europäische Ausland werden Commodore 64 mit einem »B-Board« vertrieben. Computer mit dieser Board-Version sind weder VDE- noch FTZ-zugelassen. In Deutschland darf nur die zugelassene A-Version gekauft werden. Wegen grundsätzlicher Unterschiede zwischen den einzelnen Board-Versionen können diese Computer nicht den postalischen Bedingungen entsprechend umgerüstet werden. Einige Händler scheinen sich mit diesen B-Versionen aus dem Ausland eingedeckt zu haben. Sollte Ihr C 64 den Fernsehempfang des Nachbarn stören, haben Sie vielleicht einen B-Computer erwischt, der möglicherweise zu Schwierigkeiten mit den zuständigen Behörden (Post) führen kann. Wollen Sie sich einen C 64 kaufen, so vergewissern Sie sich bitte vorher, ob dieser die A-Platine besitzt.

Schneller mit Modul

Roreger bietet die Schnelllade- und Schnellspeicher-Systeme TurboROM-Disksystem und Tapesystem auf Modul für den Expansion-Port des C 64 an. Dadurch soll kein Eingriff mehr in den Computer oder das Floppy-Laufwerk erforderlich sein. Außerdem soll das System voll kompatibel zu Steckplatzerweiterungen sein. Das Tape-System arbeitet auch mit Simons Basic und Files, die mit Turbo-Tape oder Fast-Tape aufgenommen wurden zusammen. LOAD, SAVE und VERIFY sollen bis zu 10 mal schneller, beim Disk-System bis zu 7 mal schneller (abspeichern 3,5 mal) sein.

Info: Dipl.-Ing. K. Roreger, Liebigstr. 28, 4780 Lippstadt

Floppy 1541 jetzt für PC-1500

Den Anschluß von bis zu drei 1541-Laufwerken an den Sharp PC-1500 soll das Floppy-Interface von Tramsoft ermöglichen. Alle Möglichkeiten des C-1541-DOS sollen dabei erhalten bleiben. Ausgerüstet mit zusätzlicher Software, kann am gleichen Interface auch ein Drucker oder Plotter mit Centronics-Eingang betrieben werden.

Info: SRS Ing. Rudolf W. Fankhauser, Postfach 1115, 7893 Jestetten, Tel. (04153) 62593

Software für Kleinbetriebe

Um aus dem C 64 ein leistungsfähiges Arbeitsmittel für Kleinbetriebe zu machen, hat SM Software eine eigene Software-Serie mit dem Namen »Small Business« entwickelt. Diese kaufmännisch orientierte Software-Serie beinhaltet die Programmbausteine Textverarbeitung, Lohn-/Gehaltsabrechnung, Lagerverwaltung, Adreßverwaltung sowie ein reines Fakturierungsprogramm. Alle Programme zusammen sollen zirka 1000 Mark inklusive Mehrwertsteuer kosten.

Info: SM Software AG, Small Business Service, Scherbaumstr. 33, 8000 München 83, Tel. (089) 6371211

Ferien mit dem Computer

Für Einsteiger, Fortgeschrittene und Computerfüchse bietet sich durch fun & future in den Sommerferien die Möglichkeit, zwei Wochen Computerurlaub in Bad Harzburg zu verbringen. Das Angebot richtet sich an 11- bis 19jährige, die folgende Kurse belegen können: Logo, Basic I bis III, Spiele, Maschinensprache und Pascal. Der täglich 4stündige Computerunterricht ist kombiniert mit einem vielfältigen Freizeitangebot.

Info: fun & future Schopka KG, Mittelstr. 86, 2000 Norderstedt, Tel. (040) 5243176

Data-Com Plus — Datasette leicht justiert

Nie mehr LOAD ERROR! Damit dieser Traum zur Wirklichkeit wird, muß man eigentlich nur den Tonkopf der Datasette neu justieren. Dabei soll das Gerät Data-Com Plus, das zusammen mit einer Einmeßkassette zum Preis von 39 Mark erhältlich ist, helfen.

Info: Computer Store, Herzebrockerstr. 46, 4830 Gütersloh 1, Tel. (05241) 12080

Jetzt auch in Deutschland ein Strategiespiele-Versand

Für alle, die sich für Strategiespiele begeistern, wurde jetzt ein Spezialversand für diese Spielgattung gegründet. Die Firma Thomas Müller Computer-Service hat sich auf den Vertrieb von SSI- und Avalon-Hill-Strategie-Software spezialisiert. Sie besitzt außerdem die Rechte, deutsche Anleitungen für SSI-Spiele herzustellen. Raubkopierer können dort allerdings nichts holen: Die deutschen Anleitungen werden nur zusammen mit der Software verkauft. Das Spiel Cosmic Balance, das in unserem Artikel über Strategiespiele für Herbst dieses Jahres angekündigt wurde, dürfte bei Erscheinen dieser Ausgabe schon lieferbar sein.

Info: T. Müller Computer-Service, Postfach 2526, 7600 Offenburg, Tel. (0781) 72004

Speichererweiterung für C 16

Auf insgesamt 32 KByte RAM läßt sich der C 16 mit dem 16-KByte-Modul der Firma Jeschke erweitern. Das Modul (Preis 119 Mark) wird einfach in den Expansions-Port des C 16 eingesteckt und ist sofort betriebsbereit. Im Grafikmodus wird damit der zur Verfügung stehende Speicher von mageren 2 auf immerhin 18 KByte aufgestockt, so daß endlich sinnvoll mit Shapes gearbeitet werden kann. (ev)

Info: Klaus Jeschke, Im Birkenfeld 3, 6233 Kelkheim

Neues Oxford-Pascal für den C 64

Eine neue, verbesserte Version des Oxford-Pascal-Compilers (Test in Ausgabe 12/84) wird seit Anfang April von der Firma CPL angeboten. Besitzer der alten Version können den neuen Compiler im Rahmen einer Umtauschaktion gegen einen geringen Unkostenbeitrag erhalten. Neben der Diskettenausgabe ist Oxford-Pascal jetzt auch in einer Kassettenversion erhältlich.

Info: CPL Computer plus Soft GmbH, Bahnstr. 20-26, 4220 Dinslaken.

»60000 Bytes free« beim C 16

Die 64-KByte-Erweiterung von Kingsoft macht's möglich: über 60 KByte stehen beim C 16 für Basic-Programme zur Verfügung (zum Vergleich: 38 KByte beim C 64). Auch bei Einsatz der hochauflösenden Grafik bleibt mit gut 50 KByte immer noch reichlich Platz fürs Programm. Die Erweiterungsplatine für 199 Mark wird ohne Löten direkt ins C 16-Gehäuse eingebaut; der Steckmodulport bleibt frei.

Info: Kingsoft, Fritz Schäfer, Schnackebusch 4, 5106 Roetgen

Eureka — Das 85000 Mark-Adventure

Für die Lösung des Adventure-Paketes Eureka kann man 85000 Mark bekommen. Eureka besteht aus 5 einzelnen Abenteuerspielen, die alle in sich abgeschlossen sind. Die einzelnen Lösungen ergeben zusammengesetzt eine Telefonnummer in England. Der erste Anrufer dieser Nummer erhält die Siegprämie. Das Original des Spieles ist in Englisch geschrieben. Zum ersten mal gibt es parallel aber eine **deutsche Version**, die ab dem 20. März erhältlich sein soll. Unter der Hotline-Telefonnummer kann der Abenteuer-Freund jederzeit erfahren, ob der Preis schon vergeben ist. Sollte das »Eureka-Rätsel« bis zum 31. Dezember dieses Jahres nicht gelöst sein, wird der Preis unter den Einsendern einer mitgelieferten Postkarte aufgeteilt.

Info: LINEL Handelsfirma, Landquartstr. 46 A, CH-9320 Arbon, Schweiz

RS232-Datenübertragung

Märki und Lenz bieten ein bidirektionales RS232C-(V.24-)Interface an, das an den IEC-Bus des C 64 angeschlossen wird. Angesprochen wird es wie ein Drucker über die Geräteadresse 4. Mit der Sekundäradresse wird die Art der Codewandlung eingestellt. »1« öffnet den Linearkanal, »2« schaltet die CMB/ASCII-Code-Wandlung ein. Mit kurzen Befehlen können dabei bis zu 16 Zeichen umdefiniert werden.

Das Interface arbeitet RS232-seitig mit 7 Bit, keine Parität, 1 Stop-Bit und einstellbarer Baudrate (50 bis 9600 Baud).

Der Einsatzbereich des Interfaces liegt in der Ansteuerung von RS232-Druckern und der Datenübertragung zwischen zwei Computern. Mit dem Interface lassen sich beispielsweise Vizawrite-Textdateien an einen IBM-PC übertragen und dort mit Wordstar weiterbearbeiten. Als Software wird dazu ein Viza-Konvertierungsprogramm für den C 64 und ein Terminalprogramm für den IBM (Cross-Talk, Open Access) benötigt.

Info: Märki und Lenz, Bernstr. 15, CH-3114 Wichtrach, Tel. (0041/31-982152, Preis: 454 Mark



Die Kuriositäten-Ecke

Interessantes, Seltsames und sonstige Ungereimtheiten aus dem Computer-Dschungel. Diesmal beschäftigen wir uns mit unbeabsichtigten, beabsichtigten und folgenreichen Ausrutschern verschiedener Computerfirmen.

— Steven Spielberg verkaufte die Rechte zu E.T. für 22 Millionen Dollar an Atari. Atari brachte das Spiel allerdings nur für das VCS-Telespiel heraus —, und dort war es ein totaler Flop. Trotzdem entschloß sich Atari, die Rechte zu »Gremlins« ebenfalls zu kaufen, und zwar für einen Betrag in etwa derselben Größenordnung. Diesmal aber wurde die weise Entscheidung getroffen, das Spiel für Homecomputer herauszubringen. Die Kritiken mehrerer amerikanischer Fachzeitschriften zu diesem Spiel waren allerdings schon wieder vernichtend. Ob Atari in diesem Fall wieder so viel Pech haben wird, bleibt abzuwarten.

— Ein weiterer »Fehlgriff« von Atari: Alan Alda, ein bekannter amerikanischer Fernsehstar, unterzeichnete einen zehn Millionen Dollar Vertrag mit Atari, um für eine Dauer von fünf Jahren deren Computer in der Werbung anzupreisen. Nachdem Atari an Jack Tramiel verkauft wurde, fühlte Alda sich nicht mehr gebunden und sagte in einer Talkshow etwas über »Computer, die vom Markt verschwinden könnten«. Atari kann nichts gegen solche Äußerungen tun, der Vertrag wurde nämlich noch mit dem ehemaligen Eigentümer, der Warner Communications Company, abgeschlossen.

— Apple ist mittlerweile groß ins Geschäft mit T-Shirts, Mützen und Tassen eingestiegen, die alle diesen angebissenen Regenbogenapfel tragen. Sogar in der 64'er Redaktion stand schon eine Apple-Kaffeetasse (der Besitzer wurde wegen Hochverrats mit dem Verschwinden seiner Kaffeetasse bestraft). Da dieses Mailorder-Geschäft offensichtlich ganz gute Gewinne einbringt, sollte sich Commodore nicht nur auf Bayern-T-Shirts beschränken.

— Die englische Firma »Bad Taste Software« hat mit ihrem Spiel »Di's Baby« erhebliches Aufsehen in den englischen Medien erregt. In einer der fünf Spielrunden muß man als Prinz Charles mit einem Nachttopf herumrennen, um die »Reste« des Kleinen aufzusammeln — was im Lauf der Zeit immer geschmackloser wird. Im weiteren Spielverlauf muß man vorbei an verrückten Doktoren, Journalisten etc., um zu Lady Di's Schlafzimmer vorzudringen. Dort muß man dann das nächste Baby produzieren (jedes neue Baby gibt Bonuspunkte). Kommentar der englischen Zeitschrift Commodore Horizons: »A monumental exercise in bad taste« (ein gigantisches Beispiel für schlechten Geschmack). Über Geschmack läßt sich bekanntlich streiten, aber bei solcher Publicity müßte das Spiel eigentlich ganz gut verkauft werden.

— Coleco, USA, verteilte an die Käufer des Coleco-Adam umsonst die sogenannten Cabbage Patch Dolls, Puppen, die in USA ähnlichen Erfolg haben wie bei uns die Barbies. Die Produktion des Adam-Computers wurde mittlerweile eingestellt. Vielleicht hätte Coleco andersrum mehr Erfolg gehabt (zu den Puppen den Computer umsonst).

— Nachdem letztes Jahr viele Hardware-Hersteller Millionenverluste erleiden mußten, sind auch bei den Softwarefirmen Einbrüche erfolgt. Nach Sirius-Software, die sich überhaupt nicht mehr retten konnten, fiel nun auch Hesware dem Pleitegeier zum Opfer. Hes, die hauptsächlich Software für den C 64 produzierten, wurden samt Inventar und Programmierer durch die Firma »Avant Garde« ersteigert, die bisher hauptsächlich auf dem Apple II und dem Personal Computer-Markt aktiv war. Synapse Software hatte ebenfalls zu leiden, konnte sich aber gerade noch durch einen billigen Verkauf ihres gesamten Lagers und durch Lizenzverkäufe an englische Firmen (die die Synapse-Spiele auf den Sinclair Spectrum umschreiben!) retten. Synapse wird mittlerweile voll von Broderbund-Software finanziert, bleibt aber weiterhin eine unabhängige Firma. Synapse-Chef Igor Wolosenko kündigte an, daß einige gemeinsame Projekte von Synapse und Broderbund in Planung sind.

(M. Kohlen/aa)

Nachdem wir dieses Mal nur über Firmen berichtet haben, wird unsere nächste Kuriositätenecke wieder etwas bunter gemischt sein. Was zu erwarten ist, wollen wir noch nicht verraten. Wenn Sie etwas Interessantes, Kurioses oder Witziges herausfinden sollten, dann schreiben Sie uns ruhig (Ihre Informtionen sollten auf der Wahrheit beruhen und nachprüfbar sein).

Wie funktioniert Multicolor beim VC 20?

Ich habe seit einem Jahr einen VC 20 und stehe jetzt vor folgendem Problem: Ich sehe in manchen Computerspielen Figuren im Mehrfarbmodus. Und zwar sind diese Farben gezielt angewendet worden. Ich weiß zwar, daß es einen POKE-Befehl gibt, der die Zeichen farbig erscheinen läßt, jedoch ist dies dann kunterbunt gemischt. Können Sie mir sagen, wie man Multicolor gezielt anwenden kann?

Martin Knizia

Ja, Sie müssen sich nur erst einen anderen Zeichensatz in den Speicher setzen. Tun Sie das in der Grundversion durch POKE 36869,255. Der Videochip greift jetzt auf die Zeichen im Speicherbereich ab dezimal 7168 zu. Dort können Sie Ihre Zeichen ablegen. Allerdings bedeutet hier (ich hoffe Sie beherrschen das Binärsystem) nicht wie im normalen Hires-Modus die Eins einen gesetzten Punkt und die Null einen gelöschten Punkt. Um die Multicolor-Grafik anzusprechen, setzen Sie für die verschiedenen Farben Bitpaare ein. »10« steht für die Zeichenfarbe, die für jede Bildschirmstelle (7680 - 8191) in der zugeordneten Farbspeicherstelle (37888 bis 38912) steht. »01« steht für die Rahmenfarbe, die die erste Multicolorfarbe darstellt. Die zweite Multicolorfarbe, Bitkombination »11«, ist die Hilfsfarbe, die in Bit 4 bis 7 des Videoregisters 36878 liegt.

Dadurch hat man natürlich nur noch 4 mal 8 Punkte große Grafikzeichen, die Farbe läßt sich aber jetzt gezielt ansteuern.

Wesentlich ausführlicher ist das Problem bereits im VC 20-Kurs in der Ausgabe 3/85 dargestellt worden.

Probleme mit Super Line?

Ich habe das Programm SUPER LINE aus dem 64'er Sonderheft (Tips & Tricks) abgetippt und die richtigen Checksummen erhalten. Es wurde auch keine Fehlermeldung ausgegeben. Nach dem Befehl »O« und RETURN erscheint jedoch nur ein wirres Muster. Nach »F« ist der Bildschirm im normalen Zustand mit Cursor und »Ready«. Nach einem erneuten »O« und einem RUN/STOP-RESTORE erscheint ein ganzer Bildschirm voller As.

Eine Eingabe eines Zeichens mit 80 Zeichen pro Zeile ist nicht möglich. Ich habe aber angenommen, daß das mit obigem Programm möglich sein sollte.

Kurt Lüscher-Feldmann

Viele Leser haben das gleiche Problem: Sie lesen die Anleitung nicht ganz durch. In der Anleitung zu Super Line steht ganz deutlich, daß Text mit dem Befehl »W, x, y, "Text"« im 80-Zeichen-Modus **ausgegeben** werden kann. Von Eintippen im 80-Zeichenmodus war nicht die Rede. Aber früher oder später wird uns vielleicht ein Leser ein 80-Zeichen-Programm mit dieser Möglichkeit einschicken?!

Nochmals »Graphics Basic«

Zu Ihrem Artikel über »Graphics Basic« in Ausgabe 12/84 habe ich einige Fragen:

1. Wo kann man Graphics Basic beziehen und wieviel kostet diese Erweiterung genau?

2. Kann man die anderen Basic-Befehle (zum Beispiel POKEs für Listschutz) einsetzen?

3. Wieviele Farben sind mit Graphics Basic realisierbar?

4. Kann man mit dem Window-Befehl den Bildschirm in verschiedene Aktivitätszonen wie bei Dallas Quest (Oben Grafik, unten Text, der unter der Grafik verschwindet) aufteilen?

5. In wieviele Aktivitätszonen kann man den Bildschirm unterteilen?

Ulrich Reiter

Die Firma Ariolasoft, die das Programm ursprünglich vertreiben wollte, konnte sich nicht dazu entschließen, das Programm auf den deutschen Markt zu bringen. Vielleicht, weil die Herstellerfirma »HES« wegen Bankrott versteigert wurde.

Mit anderen Worten: Graphics Basic ist leider nicht mehr über den Fachhandel zu beziehen. Nach Auskunft von Ariolasoft werden lediglich noch die vorbestellten Exemplare ausgeliefert, ein späterer Verkauf ist ausgeschlossen.

Zu 2.: Man kann die anderen Basic-Befehle ohne Schwierigkeiten einsetzen.

Zu 3.: Alle

Zu 4.: Das geht mit den Befehlen »TEXT FROM x TO y« und »GRAPHICS FROM x TO y«.

Zu 5.: Mehr als drei Aktivitätszonen sind leider nicht möglich.

Aus dem Takt geraten?

Sie schrieben in Ihrer Turbo-Pascal-Story in einer älteren Ausgabe, daß der Z-80 des CP/M-Moduls mit 2 MHz getaktet ist. Ist das ein Fehler Ihrerseits, oder hat Commodore das CP/M-Modul verbessert?

Dirk Müller

Der Fehler liegt bei uns, und Commodore hat nichts verbessert (sowas kann man von Commodore erfahrungsgemäß nicht erwarten). Die Produktion des CP/M-Moduls wurde schon vor einiger Zeit eingestellt. Es ist daher ratsam, alles was mit CP/M und dem C 64 zu tun hat am besten zu vergessen.

Das CP/M-Modul soll unbestätigten Berichten zufolge außerdem nicht mit den neuesten C 64-Versionen zusammenarbeiten. Interessant wird das Thema CP/M allerdings wieder mit dem C 128, der einen Z-80-Prozessor mit 4 MHz Taktfrequenz fest eingebaut hat.



ROM-Listings gesucht

Wo kann man Schaltpläne zur Floppy 1541 und zum Drucker MPS-802 erhalten? Bietet jemand ein kommentiertes ROM-Listing des MPS-802-Betriebssystems an? Gibt es irgendwo kommentierte Listings von Simons Basic, Exbasic, Assemblern etc.?

Manfred Grebler

Schaltpläne zu Commodore-Geräten kann Ihnen nur Commodore selbst beziehungsweise Ihr Commodore-Händler beschaffen. Ein kommentiertes Listing von Simons Basic finden Sie beispielsweise im Commodore 64-Buch, Band 5, »Ein Leitfaden durch Simons Basic« (Markt & Technik).

Hi-Eddi mit C 1525?

Das Programm Hi-Eddi (Ausgabe 1/85) ist einfach toll. Aber ich habe ein Problem. Das Hi-Print-Programm läuft nicht auf meinem Grafik-Drucker C 1525. Alle Versuche, es entsprechend abzuändern, sind bei meinen geringen Programmierkenntnissen fehlgeschlagen.

Andreas Neuner

In der nächsten 64'er finden Sie Druckerroutinen für MPS 801 und 802.

Vizawrite-Hilfe gesucht

Seit einiger Zeit verarbeite ich meine Texte mit einer englischen Version von Vizawrite. Mit ein paar Kunstgriffen läßt sich dieses Programm ganz gut handhaben. Es gibt aber einen Schönheitsfehler: Auf dem Bildschirm erscheinen nicht die deutschen Umlaute, und meine bisherigen Versuche, selbst Bildschirmzeichen zu definieren, schlugen fehl, da diese von Vizawrite wieder überschrieben werden. Wer kennt eine nicht allzu komplizierte Lösung für mein Problem?

Bertram Hafner

Mitglieder gesucht

Der VC 20 User Club »Byte Sprinter« sucht noch Mitglieder.

Manfred Beier

Info: VC 20 User Club »Byte Sprinter«, Manfred Beier, Narzissenweg 3, 4044 Karst 1.

8-Zoll-Floppy für C 64?

Ich benötige für meinen C 64 einen Massenspeicher mit wesentlich höherer Kapazität als sie die 1541 bietet.

Joachim Kaluza

Eine Möglichkeit wäre die SFD 1001 von Commodore. Sie hat eine Speicherkapazität von rund 1000 KByte. Zusätzlich ist eine IEEE 488-Schnittstelle erforderlich (siehe 64'er, 3/85).

CBM-Gehäuse gesucht

Ich möchte meinen C 64 wie in der 64'er Ausgabe 8/84 beschrieben in ein CBM-Gehäuse der 3000/4000er Serie einbauen, konnte aber bislang immer noch keins bekommen. Welcher Leser kann mir helfen?

Jörg Stegemann

TA Gabriele 8008 am C 64

Ich suche ein Interface zum Anschluß der Typenradschreibmaschine TA Gabriele 8008 an den C 64. Ausgabe 3/84

Oskar Greifenberger jun.

Dieser Brief wurde mit einer TA 8008 in Verbindung mit dem »Textomat« von Data Becker geschrieben. Damit dürfte die Frage von Herrn Greifenberger jun. beantwortet sein. Das Interface gibt es bei der Firma Daum Elektronik, Kagenhof 3, 8501 Veitsbronn. Die Typenbezeichnung lautet: Schnittstelle VC 8008. Ich habe die Schnittstelle im November 1984 für 289 Mark gekauft.

Dieter Seifried

Welches Betriebssystem?

Wie kann ich feststellen, ob mein 1526-Drucker mit dem alten oder mit dem neuen Betriebssystem ausgerüstet ist? Ausgabe 3/85. *Günter Reuter*

Welches Betriebssystem eingesetzt ist, kann man sehr einfach feststellen, indem man einen Drucker-Selbsttest durchführt (Gerät mit festgehaltener Papiervorschub-Taste einschalten). In der ersten gedruckten Zeile erscheint die Bezeichnung »REV.« mit einer Zahl dahinter. Ist diese Zahl »1.0«, dann ist das alte Betriebssystem eingesetzt; bei »7.0« ist es das neue.

Ob der Drucker im 1525- oder im 1526-Modus arbeitet, kann man feststellen, wenn man ihm einen Text über die Sekundäradresse 7 schickt. Wird der Text gedruckt, so ist der Drucker im 1525-Modus, andernfalls im 1526-Modus. Ist der Drucker in der 1525-Betriebsart, dann kann man ihn bei neuem Betriebssystem wie folgt auf den 1526-Modus umstellen:

1. Gerät öffnen (Achtung, Garantiefrist beachten)
2. Das IC mit der Bezeichnung »U4D« suchen (steht auf der Leiterplatte neben dem IC)
3. Pin 16 des IC auf Masse legen

Der Drucker arbeitet jetzt im 1526-Modus. Hendrik Hartje

Fragen Sie doch

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der Karte »Lesermeinung«). Wir veranlassen, daß sie von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht, die übrigen brieflich beantwortet.

Laden mit Bild?

Während der C 64 ein Programm von Kassette lädt, zeigt er normalerweise kein Bild auf dem Monitor an. Nun habe ich aber an einigen Maschinensprache-Programmen gesehen, daß es doch möglich ist, ein Bild während des Ladens stehen zu lassen. Ist dieser Effekt auch in Basic zu erzielen oder ist dazu die Kenntnis von Maschinensprache notwendig?

Dieter Kurbjuhn

Vom Basic aus ist das nicht möglich. Da der Videochip und der Prozessor abwechselnd auf den selben Datenbus zugreifen, kann es bei Ladevorgängen von Kassette zu Zeitproblemen kommen, die das System zum Absturz bringen könnten. Um das zu verhindern, schaltet das Betriebssystem während der Dauer des Kassettenzugriffs den Videochip einfach aus, wodurch natürlich das Bild verschwindet.

Um das zu verhindern, darf man nicht auf die vorhandenen Betriebssystemroutinen zugreifen — wie es das Basic automatisch macht, sondern muß sich eigene Laderoutinen schreiben. Das allerdings ist nur in Maschinensprache möglich.

Assembler für »Illegals«?

Ich habe von einigen Maschinenbefehlen gehört, die die CPU 65xx versteht, obwohl sie in keinem Handbuch zu finden sind. Hat jemand nähere Informationen darüber oder gibt es schon einen Assembler dafür?

Klaus Heinz

In der Ausgabe 3/84 des 64'er-Magazins haben wir eine Beschreibung dieser illegalen Opcodes gebracht. Weitere Beiträge zu diesem Thema sind in Vorbereitung.

Bei der Anwendung dieser vom CPU-Hersteller nicht unterstützten Opcodes ist aber Vorsicht geboten: Während die »legalen« Befehle unter Garantie auch bei späteren Prozessorversionen funktionieren, ist das bei den »illegalen« nicht unbedingt der Fall. Unter Verwendung solcher Codes geschriebene Maschinenprogramme sind daher nicht immer auf Prozessoren der gleichen Familie (zum Beispiel vom 6502 zum 8502 des C 128) übertragbar.

Wo gibt's ausländische Computermagazine?

Ich würde gerne auch ausländische Computermagazine lesen. Deshalb meine Frage: Wie beziehungsweise wo bekommt man englischsprachige Computermagazine?

Roland Rechinger

Englische und amerikanische Zeitschriften gibt es in jeder größeren Stadt im Bahnhofsbuchhandel zu kaufen. Manche dieser Läden erhalten diese Zeitschriften allerdings oft mit erheblicher Verspätung (manchmal bleiben auch die Lieferungen aus den USA ganz aus). Wenn Sie also an einem ständigen Konsum dieser Blätter interessiert sind, sollten Sie ein Abo in den USA oder England bestellen. Informationsreiche Zeitschriften sind Compute!, Computes Gazette, Computer Games, Electronic Games, Personal Software (USA) sowie Commodore Horizons, Your Computer, Computer & Video-Games (England).

Copyright-Problem

Darf ich eigentlich Programmteile, die im 64'er abgedruckt sind, für meine eigenen Programme benutzen und diese dann wieder einschicken?

Rene Frehner

Da der Verlag Markt & Technik das Copyright an diesen Programmen besitzt, dürfen Sie auch einzelne Programmteile nur verwenden, wenn das schriftliche Einverständnis von Markt & Technik vorliegt, falls Sie Ihr Programm einem anderen Verlag anbieten.

Schicken Sie Ihr Listing aber an uns ein, dürfen Sie dieses Einverständnis natürlich immer voraussetzen.

Generell dürfen Sie einem Verlag Listings nur dann anbieten, wenn alle Rechte an dem Programm bei Ihnen liegen, das heißt in der Regel, wenn Sie es selbst geschrieben haben. Falls das Programm nur eine umgeschriebene Version eines anderen ist, oder falls einzelne Routinen aus anderen Quellen stammen, müssen Sie das unbedingt im Anschreiben erwähnen, sonst kann das unangenehme Folgen haben.

Falls Sie das Programm jedoch ausschließlich für Ihren privaten Gebrauch verwenden, dann brauchen Sie sich um solche Dinge keine Gedanken zu machen. Es entspricht aber einer guten Programmierer-Tradition und auch der allgemeinen Fairneß, bei aus anderen Programmen entnommenen Routinen die Original-Quelle und den Namen des Autoren der Routine anzugeben.

Modulbereich nutzen?

Wie kann man beim VC 20 mit 24 KByte den geänderten Zeichensatz in den Modulbereich ($A000-$BFFF) verlegen ? Welcher Wert muß dazu in die Speicherstelle 36869 gePOKEt werden ? Muß dazu auch der Bildschirmspeicher verlegt werden? *Oliver Berger*

Die Bits 0 bis 3 aus Adresse 36869 (Register 5 des VIC) bestimmen die Lage des Zeichengenerators im Speicher, und zwar nach einem recht komplizierten Schema, bei dem einige Bits der generierten effektiven Startadresse des Zeichenspeichers immer auf Null sind. Leider gehören auch Bit 13 und 14 der Zeichenspeicher-Adresse zu diesen Null-Bits. Dadurch ist der Speicherbereich ab $A000 (= 1010 0000 0000 0000 binär) nicht als Zeichengenerator-Adresse einstellbar. Auch der Bildschirmspeicher läßt sich beim besten Willen nicht in diesen Adreßbereich verlegen. In der Ausgabe 1/85 des 64'er-Magazins finden Sie im VC 20-Kurs eine ausführliche Beschreibung dieser Problematik.

Hier gibt's Lightpens

Wo bekomme ich einen anschlußfertigen Lightpen für den C 64? Ausgabe 2/85

Markus Lucassen

Lightpens mit Demonstrations-Software in Simons Basic gibt's bei mir für 50 Mark auf Bestellung. Anwendungsprogramme für das Zeichnen auf dem Bildschirm sind in Vorbereitung.

Werner Backes

Info: Werner Backes, Talstr. 15, 6695 Tholey

Die Firma Softline bietet den »Tech-Sketch«-Lightpen in verschiedenen Versionen ab 159 Mark an, von Stack Computer Services gibt es den »Stack«-Lightpen. Beide Lightpens werden inklusive komfortabler Grafik-Software geliefert. In unserem Testbericht in der Ausgabe 4/85 können Sie sich genauer über den Tech-Sketch informieren.

Info: Softline, R. Alverdes, Schwarzwaldstr. 8a, 7602 Oberkirch
Stack Computer Services Limited, 290-298 Derby Road, Bootle, Merseyside, England.

Wollen Sie antworten?

Wir veröffentlichen auf dieser Seite auch Fragen, die sich nicht ohne weiteres anhand eines guten Archivs oder aufgrund der Sachkunde eines Herstellers beziehungsweise Programmierers beantworten lassen. Das ist vor allem der Fall, wenn es um bestimmte Erfahrungen geht oder um die Suche nach speziellen Programmen. Wenn Sie eine Antwort auf eine hier veröffentlichte Frage wissen — oder eine andere, bessere Antwort als die hier gelesene, dann schreiben Sie uns. Antworten publizieren wir in einer der nächsten Ausgaben. Bei Bedarf stellen wir auch den Kontakt zwischen Lesern her.

Fernschreiber am C 64?

Wer hat eine Schaltung, um einen C 64 an einen Fernschreiber anzuschließen? Wer kennt Literatur, in der dies beschrieben wird? Am besten wäre eine Schaltung, mit der man auch noch Lochstreifen vom Fernschreiber in den Rechner einlesen kann. Bernd Alef

C 64-Speicher puffern?

Ich möchte den RAM-Speicher des C 64 puffern, um Datenverluste beim Abschalten zu vermeiden. Wer kann mir sagen, wie das zu bewerkstelligen ist? Erwin Rieks

Bisher haben wir leider keinen Schaltplan oder eine Bezugsquelle für eine derartige Schaltung gefunden. Eine Firma aus den USA, die bereits für den Apple II eine 64-KByte-Karte anbietet, auf die bei einem Reset der Inhalt des gesamten Speichers gezogen wird, will diese Karte (Name: »Wild Card«) auch für den C 64 anbieten. Wann und ob diese Erweiterung in Deutschland erhältlich ist, ist jedoch noch völlig ungewiß.

Falls Sie jedoch auch mit einem akkugepufferten Zusatzspeicher zufrieden sind, ist vielleicht das Xtend 64-Modul von Roßmöller das richtige für Sie. Dieses Modul erweitert den C 64 wahlweise um 32 KByte RAM (akkugepuffert) oder 128 KByte EPROM. EPROM und RAM können auch gemischt eingesteckt werden.

Leser fragen — Willi Brechtl antwortet

Hallo liebe Leser, hier bin ich wieder, um Eure Fragen zu beantworten.

Ich werde mich hauptsächlich um Leserbriefe kümmern, die nicht in das sachliche Einerlei des Leserforums passen. Zum Beispiel Fragen, die sich aus dem einen oder anderen Grund nur ganz subjektiv beantworten lassen. Oft genug tauchen auch Probleme auf, die sich nicht mit einem kurzen Antwortsatz abhandeln lassen. Und wenn selbst eine längere Antwort im Rahmen des Leserforums nicht mehr ausreichen würde, dann ist das ganz klar ein Fall für Willi Brechtl.

Also: Wenn Sie als Anfänger Probleme mit Computer, Software oder Handbuch haben, dann wenden Sie sich in Zukunft doch einfach vertrauensvoll direkt an mich.

Fragen zum C 16

Ich habe mir erst kürzlich den C 16 zugelegt, und habe zu diesem einige Fragen:

1. Mir ist aufgefallen, daß die Speicherkapazität des C 16 unter Verwendung von hochauflösender Grafik um rund 10 KBytes verringert wird. Woran liegt das und wie kann man es verhindern?

2. Der C 16 hat eine recht geringe Speicherkapazität. Gibt es eine Speichererweiterung?

3. Zur Zeit gibt es noch sehr wenig Software und Literatur zum C 16. Werden im 64'er künftig auch Artikel über den C 16 veröffentlicht? Wird Data Becker Bücher für den C 16 veröffentlichen?

4. Da Joysticks und Datasette 1530 nicht kompatibel zum C 16 sind, hätte ich gerne gewußt, ob es Interfaces oder Stecker zum Anschluß der Datasette 1530 und der »alten« Joysticks geben wird? Andre Bremer

Zu 1. Da die Grafik ja auch in irgendeinem Speicherbereich liegen muß, ist es logisch, daß dieser Platz, wenn schon kein eigener Grafikspeicher vorhanden ist, das Basic-Ram benützt und verkleinert. Dabei werden 8 KByte für die Hires-Grafik und 2 KByte für den Farbspeicher verbraucht. Verhindern kann man das nur dadurch, daß man einfach keine hochauflösende Grafik benutzt.

Zu 2. Eine 16-KByte-Speichererweiterung wird derzeit von der Firma Jeschke angeboten. Eine 64-KByte-Erweiterung ist über Kingsoft zu beziehen. Die Firma Roßmöller entwickelt derzeit eine 32-KByte-Erweiterung mit zusätzlichen EPROM-Steckplätzen.

Zu 3. und 4. Data Becker müssen Sie schon selbst fragen. Im 64'er wird der C 16 entsprechend seiner Verbreitung auf dem Markt berücksichtigt. Einen Selbstbau-Stecker zum Anschluß der »alten« Datasette an den C 16 haben wir bereits gebracht, den entsprechenden Adapter für Joysticks finden Sie als Bauanleitung in dieser Ausgabe.

Sie sehen also, wir haben die C 16-Besitzer nicht vergessen und werden auch in Zukunft am Ball bleiben — wie bei allen aktuellen Commodore-Computern.

Info: Roßmöller GmbH, Finkenweg 1, 5309 Meckenheim
Klaus Jeschke Hard & Software, Im Birkenfeld 3e, 6233 Kelkheim
Kingsoft, Fritz Schäfer, Schnackebusch 4, 5106 Roetgen

»Load Error« bei Datasette?

Ich habe folgendes Problem mit meiner Datasette: Sie macht andauernd »? LOAD ERROR« trotz Kassettenwechsel. Mit meiner Datasette ist alles in Ordnung. Man hat mir nun gesagt, daß vielleicht etwas mit meinem Mikroprozessor im C 64 nicht stimmt. Kann es tatsächlich daran liegen? Ronny Gaab

Leider kann man bei so allgemeinen Angaben nur vage Vermutungen über die Fehlerursache anstellen. Tritt der Lesefehler nur bei fremden Programmen auf oder auch bei selbst abgespeicherten? Wurde die Abspeicherung mit VERIFY überprüft, und mit welchem Ergebnis? Können nur vereinzelt Programme nicht gelesen werden oder funktioniert gar nichts mehr?

Jedenfalls ist es sehr unwahrscheinlich, daß der Fehler am C 64 liegt (an der CPU kann es schon gar nicht liegen, wenn der Computer sonst einwandfrei funktioniert).

Die häufigste Fehlerursache ist ein verschmutzter oder verstellter Tonkopf an der Datasette. Reinigen Sie Tonkopf, Bandführung und Andruckrolle von Zeit zu Zeit mit einem in Spiritus getränkten Wattestäbchen. Die korrekte Einstellung des Tonkopfes kann Ihr Fachhändler vornehmen, wenn Sie sich selbst daranwagen wollen, sollten Sie entsprechende Fachliteratur zu Rate ziehen (zum Beispiel das Cassetten-Buch von Data Becker).

Bei der Aufzeichnung denken Sie bitte unbedingt immer an das VERIFY, daß nach jedem Abspeichern auf Kassette durchgeführt werden sollte. Verwenden Sie keine Billig-Kassetten, aber auch keine Chromdioxid oder Reineisen-(»Metall«)-Bänder. C 90 und C 120-Kassetten sollten Sie ebenfalls vermeiden. Lassen Sie am Anfang und am Ende jeder Kassette mindesten 10 bis 20 Sekunden Band frei, da diese Bandstellen durch das ständige Anschlagen beim Umspulen besonders starken mechanischen Belastungen ausgesetzt sind. Achten Sie darauf, daß das Band in der Kassette frei beweglich ist (glatte Spulenwickel), eventuell ein paar Mal vollständig vor- und zurückspulen. Beachten Sie schließlich den obersten Programmierer-Grundsatz: Von jedem wichtigen Programm eine Sicherheitskopie anfertigen.

Ordnungs-Probleme

Ich nehme an einem Fernkurs für Basic teil und erhalte jeweils ein paar Lehrbriefe mit Übungs-(programmier-)aufgaben. Seit ich nun eine Diskettenstation habe, speichere ich diese Übungsprogramme auf Diskette ab. Nun habe ich aber ein fürchterliches Durcheinander auf meinen Disketten. Daher habe ich mit meinen jetzigen Fähigkeiten ein Disketten-Verwaltungsprogramm geschrieben, mit dem ich die gewünschte Übungsaufgabe auf Anhieb finden kann. Wenn ich allerdings die Übung geladen und korrigiert habe, ist natürlich anschließend mein Verwaltungsprogramm nicht mehr im Speicher, so daß ich es immer wieder nachladen muß, was recht viel Zeit kostet (immerhin 63 Blöcke). Was kann ich dagegen tun? Gibt es einen Speicher, in dem ich mein Verwaltungsprogramm ablegen und jederzeit wieder hervorrufen kann, ohne es umständlich wieder von Diskette laden zu müssen? Wie gelange ich in diesen Speicher und wie funktioniert das Ganze? Christian Wüger

Ein solcher Speicher ist vom Betriebssystem her nicht vorgesehen, kann aber auf der Maschinenspracheebene durchaus realisiert werden. In unserem Tips & Tricks-Sonderheft haben wir das »Multi-Programm-System« abgedruckt, mit dem es möglich ist, bis zu 31 (!) Basic-Programme gleichzeitig im Speicher zu halten.

Eine viel einfachere Methode ist es jedoch, für jede Diskette eine Karteikarte mit den Programmnamen anzulegen. Diese Karte können Sie in der Diskettenhülle mit unterbringen. Wenn Sie dann noch von vorneherein für Ordnung sorgen, indem Sie jeweils logisch irgendwie zusammengehörende Programme auf einer Diskette zusammenfassen (also beispielsweise Übungsaufgaben zu Lektionen 1-5 auf Diskette 1) und dies auf dem Diskettenaufkleber vermerken, werden Sie nie Probleme mit dem Suchen nach Programmen haben und können den Computer für sinnvollere Dinge als die Verwaltung der Verwaltungssoftware einsetzen.


64ER ONLINE

Welcher Drucker ist der richtige?

Einen Drucker müßte man haben, aber welchen? Ein vielschichtiges und verwirrendes Angebot macht die Entscheidung für ein Druckermodell nicht gerade einfach. Wir stellen Ihnen die wichtigsten Typen vor und sagen Ihnen, worauf Sie beim Kauf achten müssen.

Die papierlose Gesellschaft wird es auch in absehbarer Zukunft nicht geben. Menschen möchten die Resultate ihrer Arbeit nicht nur auf der flüchtigen Anzeige eines Monitors sehen. Vielmehr wird jedem Computerbesitzer schon nach kurzer Zeit bewußt, daß ein Drucker die wohl sinnvollste Ergänzung seines Computersystems ist. Der Anwendungsbereich eines Druckers ist dabei enorm vielfältig. Im industriellen Bereich dienen sie als Ausgabedrucker in Rechenzentren, erstellen Briefe, bedrucken Flugscheine und Fahrkarten, fertigen aber auch die Originale für so komplexe Erzeugnisse wie Telefonbücher, Bedienungs- und Verkaufsunterlagen. Im häuslichen Bereich haben Drucker mittlerweile ebenfalls einen weiten Aufgabenkomplex übernommen. Sie dienen hier als Listingdrucker, zur Daten- und Textverarbeitung, fertigen Grafiken an und protokollieren Meßabläufe.

Drucker statt Schreibmaschine

Der enorme Erfolg des Commodore 64 hat eine interessante Entwicklung in Deutschlands Haushalten eingeleitet: Immer mehr Computerbesitzer bearbeiten ihren privaten Schriftverkehr nicht mehr mit der Schreibmaschine, sondern mit ihrem C 64, einem Textverarbeitungsprogramm und einem Drukker. Eine verständliche Entwicklung, denn die Vorteile sind enorm. Es darf korrigiert, umgestellt, ersetzt, neu formatiert und modifiziert werden — schlechte Zeiten für Tipp-Ex. Trotzdem ist dieser Themenkomplex nicht ohne Probleme. Den für die gewünschte Aufgabe richtigen Drucker zu finden ist, in Anbetracht der Typen- und Modellvielfalt, ein umfangreiches Unterfangen.

Druckverfahren

Erstes Unterscheidungsmerkmal eines Druckers ist sein Druckverfahren. Man unterscheidet zwischen mechanischen (impact) und nichtmechanischen (nonimpact) Druckern. Zu den mechanischen Druckern zählen sowohl Drucker mit festen Typen (auf Kugelkopf, Hammer oder Rad), als auch Matrixdrucker, die ein zu druckendes Zeichen aus Einzelpunkten zusammensetzen. Nichtmechanische, anschlagfreie Drucker sind beispielsweise Laserdrucker, Thermodrucker oder Tintenstrahldrucker. Ein wesentlicher Vorteil von mechanischen Druckern gegenüber anschlagsfreien Druckern ist die Möglichkeit, direkt Durchschläge herzustellen.

Vom Arbeitsprinzip her lassen sich die Drucker in serielle Drucker, die Zeichen für Zeichen nacheinander auf das Papier bringen (Matrixdrucker) und Parallel- oder Zeilendrucker, die eine ganze Zeile auf einmal erstellen, unterteilen. Im Heimbereich haben sich in den letzten Jahren im wesentlichen nur vier verschiedene Druckertypen durchgesetzt: Die Matrixdrucker dank ihrer Flexibilität, die Typenraddrucker mit ihrem schönen Schriftbild, die Thermodrucker, weil sie preiswert und leise sind und seit neuestem auch die Tintenstrahldrucker, die viele Vorteile in sich vereinigen. Leider kann man mit Thermo- und Tintenstrahldruckern keine Durchschläge erzeugen.

Die Matrixdrucker

Seriell arbeitende mechanische Matrixdrucker drucken Zeichen als eine matrixförmige Anordnung von Punkten (Bild 1). Der Druckkopf besteht aus einer senkrecht zur Druckzeile stehenden Reihe von dünnen Rohren, die je eine Nadel enthalten. Die Nadeln werden je nach Form des zu druckenden Zeichens von Elektromagneten angestoßen und bringen über das Farbband einen Punkt auf das Papier (Bild 2). Der Nadelkopf bewegt sich dabei entlang der Zeile und druckt die Zeichen entsprechend der im Drucker gespeicherten Informationen. Die diesen Zeichen zugrundeliegende Zeichenmatrix hängt vom verwendeten Druckermodell ab. Viele Drucker verfügen sogar über mehrere solcher Zeichensätze. Die Qualität des resultierenden Schriftbildes ist direkt von der verwendeten Zeichenmatrix und indirekt von der Nadelzahl abhängig. Es leuchtet ein, daß beispielsweise eine senkrechte Linie mit einer bestimmten Länge aus 9 Punkten zusammengesetzt, wesentlich besser aussieht, als eine gleich lange aus 5 Punkten.

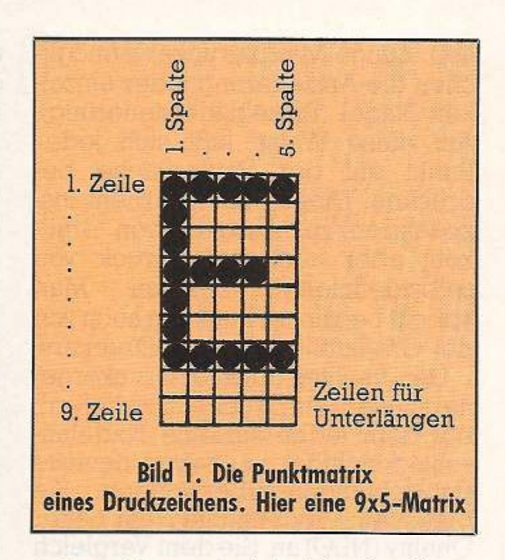


Bild 1. Die Punktmatrix eines Druckzeichens. Hier eine 9x5-Matrix

Mit verschiedenen Maßnahmen versuchen die Druckerhersteller das Schriftbild ihrer Modelle zu verbessern. So gibt es Drucker mit zwei hintereinander angeordneten Nadelreihen, damit auch die Zwischenräume zwischen den Punkten mit Druckfarbe gefüllt werden können. Ein anderes Verfahren läßt eine Zeile zweimal drucken, wobei beim zweiten Druckvorgang der Kopf zum Ausfüllen der Zwischenräume etwas versetzt wird. Solche Maßnahmen gehen natürlich auf Kosten der Geschwindigkeit. Die Druckgeschwindigkeit eines Matrixdruckers hängt hauptsächlich von der Geschwindigkeit der auslösenden Elektromagneten und der Anzahl der zu betätigenden Nadeln ab. Druckgeschwindigkeit zwischen etwa 60 und 180 Zeichen pro Sekunde sind üblich; es gibt aber auch Matrixdrucker, die mit bis zu 600 Zeichen pro Sekunde arbeiten. Die meisten der modernen Matrixdrucker arbeiten mit einer Druckwegoptimierung, bei der freie Flächen auch in der Druckzeile selbst übersprungen werden. Mit der Druckwegoptimierung läßt sich die Druckgeschwindigkeit, je nach Anwendungsfall, beträchtlich erhö-

hen. Einige Matrixdrucker ermöglichen die Ansteuerung jeder einzelnen Nadel (Einzelnadelsteuerung). Auf diese Weise läßt sich jeder Punkt auf der Papierfläche bedrucken. Diese Funktion dient insbesonders zur Ausgabe von Grafiken, aber auch zum Druck von selbstdefinierten Zeichen. Man spricht bei dieser Funktion auch von der Grafikfähigkeit eines Druckers.

Den Nachteil früherer Druckergenerationen, ein Druckbild zu liefern, bei dem jeder einzelne Nadelanschlag sichtbar ist, haben neueste Entwicklungen überwunden. Sie bieten eine sogenannte Near Letter Quality (NLQ) an, die dem Vergleich mit dem Schriftbild einer Schreibmaschine durchaus standhält (Bild 3).

Die Typenraddrucker

Manchen sicher schon von seiner Schreibmaschine her bekannt ist das Typenradprinzip. Die Drucktypen sind auf elastischen Armen (Speichen) eines Typenrades aus Metall oder Kunststoff befestigt. Durch Drehung wird das Rad in die richtige Druckposition gebracht. Dann schlägt ein Hammer die Typen gegen Farbband und Papier.

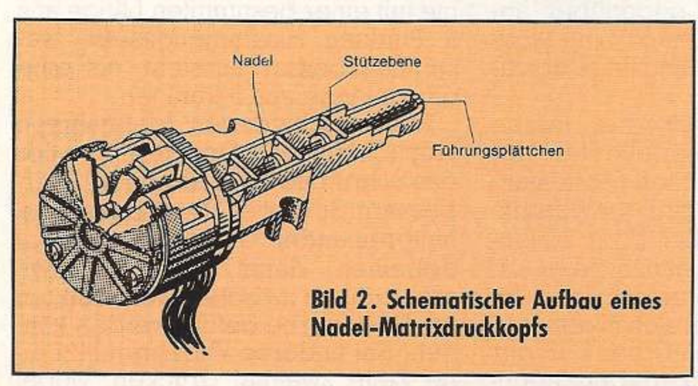


Bild 2. Schematischer Aufbau eines Nadel-Matrixdruckkopfs

Mit der NLQ-Schrift (Near Letter Quality) besitzen die Star-Drucker SG-10, SD-10 und SR-10 einen bisher unerreichten Qualitätsstandard. Die NLQ-Schrift ist durchaus mit der Qualität eines Typenraddruckers zu vergleichen, allerdings bleiben die Vorteile eines Matrixdruckers ganz erhalten.

Bild 3. Mit der NLQ-Schriftart reichen Nadel-Matrixdrucker an die Qualität von Typenraddruckern heran.

Obwohl in der Zwischenzeit auch die Matrixdrucker über gute Druckqualitäten verfügen, ist das Schriftbild eines guten Typenraddruckers bisher unerreicht. Die Domäne der Typenraddrucker liegt demzufolge auch in der Textverarbeitung. Dabei sind solche Drucker relativ langsam. Preiswerte Modelle arbeiten mit Druckgeschwindigkeiten um 20 Zeichen pro Sekunde, die professionellen Geräte schaffen teilweise über hundert Zeichen in der Sekunde. Auch bei den Typenraddruckern läßt sich die Druckqualität noch durch verschiedene Maßnahmen verbessern. Eine solche Maßnahme ist der Doppeldruck, bei dem jedes Zeichen zweimal angeschlagen wird, oder die Schattenschrift, die jedes Zeichen zweimal, aber etwas versetzt, druckt. Auch die Typenraddrucker gehören der unüberhörbaren Kategorie an. Trotzdem gibt es Unterschiede, die hauptsächlich auf die Bauart (Dämpfung, Rollenmaterial) zurückzuführen sind. Leider beherrschen die Typenraddrucker keine Grafik wie die Matrixdrucker. Sie sind vielmehr auf die Zeichen ihres Typenrades begrenzt und können keine beliebigen Punktmuster auf dem Papier drucken. Einfache Grafiken mit diesen Zeichen sind dafür nur ein unzureichender Ersatz. Größter Vorteil, neben dem Schriftbild, ist die Vielfalt der verfügbaren Schriften. Durch einfaches Wechseln des Typenrades steht ein völlig neuer Zeichensatz zur Verfügung. Andere Sprachen, wissenschaftliche Zeichen und viele Sonderzeichen stehen für die meisten Typenraddrucker zur Auswahl. Eine Sonderform des Typenraddruckers ist die Schreibmaschine mit eingebauter Schnittstelle zum Computer. Im Gegensatz zum Typenraddrucker hat die Schreibmaschine ein eigenes Tastenfeld, über das sie, auch ohne Computer, bedient werden kann. Diesen Vorteil muß man allerdings, bei gleichem Leistungsniveau auch mitbezahlen.

Die Thermodrucker

Diese anschlagsfreien Drucker verwenden ein wärmeempfindliches Spezialpapier, das beispielsweise mit einer Wachsschicht überzogen sein kann. Der Druckkopf besteht aus einer Matrix von Widerständen, die heute üblicherweise auf einem Siliziumchip integriert sind. Dieser Druckkopf ist in ständigem Kontakt mit dem Papier. Zum Drucken werden die Widerstände der Matrix selektiv erhitzt und schmelzen dabei die Oberfläche des Spezialpapiers weg, so daß die darunter befindliche Farbe als Punkt sichtbar wird. Ein anderes Prinzip arbeitet mit normalem Papier, bei dem die wärmeempfindliche Schicht auf dem Farbband aufgebracht ist. Durch Erhitzen des Widerstandes löst sich diese Schicht vom Farbband und bleibt auf dem Papier haften. Thermodrucker sind relativ preisgünstige und vor allem sehr leise Drucker. Leider stehen die Unterhaltskosten im umgekehrten Verhältnis zu den Anschaffungskosten. Sowohl das Spezialpapier als auch das wärmeempfindliche Farbband sind ziemlich teuer. Die Grafikfähigkeiten der Thermodrucker sind, durch das Matrixprinzip, durchaus gut.

Die Tintenstrahldrucker

Auch Tintenstrahldrucker arbeiten nach dem Matrix-Verfahren, doch läßt sich mit ihnen ein Zeichenfeld wesentlich feiner rastern, als mit mechanischen Matrix-Druckern: Auf eine Länge von etwa einem Millimeter passen mehr als zehn Farbtröpfchen; mit einer zusätzlichen Überlappung der Punkte erreicht man ein Druckbild, das sich von dem einer Schreibmaschine kaum mehr unterscheidet. Es gibt verschiedene Versionen von Tintenstrahldruckern. In einer Ausführung sind, wie beim Nadeldrucker, mehrere Röhrchen senkrecht übereinander angeordnet, durch die Tinte auf das Papier gelangt. Bei anderen Verfahren wird ein aus einer Düse austretender Tintenstrahl, durch elektrische oder magnetische Felder auf die entsprechende Stelle des Papiers abgelenkt. Ein besonderer Vorteil der Tintenstrahldrucker ist ihr geringer Geräuschpegel. Die Druckgeschwindigkeit mancher Tintenstrahldrucker reicht bis über 300 Zeichen pro Sekunde. Eine Chance, die Tintenstrahldrucker gegenüber anderen Druckverfahren für die Zukunft bieten, ist der Mehrfarbendruck.

Druckerauswahl — ein Entscheidungsproblem

Nun kennen Sie die wesentlichsten Druckerarten. Damit ist aber bei weitem noch nicht geklärt, welcher Drucker für welchen Zweck der geeignetste ist. Stellen wir uns zunächst den idealen Drucker vor: Er wäre so leise wie ein Tintenstrahl-

drucker, grafikfähig und schnell wie ein Matrixdrucker, hätte das Schriftbild eines Typenraddruckers, könnte Durchschläge anfertigen und hätte den Preis eines Thermodruckers. Leider gibt es dieses Gerät nicht. Die Entscheidung für einen bestimmten Drucker ist also immer ein Kompromiß, bei dem die gestellten Anforderungen am besten erfüllt werden. In Tabelle 1 haben wir für Sie einige Merkmale zusammengestellt, anhand derer Sie sich das Anforderungsprofil Ihres Druckers zusammenstellen können. Die dort verwendeten Begriffe wollen wir hier zunächst erläutern:

— Art der Schnittstelle

Es gibt Drucker, die direkt an den seriellen Bus des C 64 beziehungsweise VC 20 anschließbar sind. Zu diesen Druckern gehören selbstverständlich die Commodore-Drucker. Einige andere Druckerhersteller bieten mittlerweile ebenfalls Drucker mit dieser Schnittstelle an. Diese Drucker sind in der Regel auch in ihren Steuerungsbefehlen an das Commodore-Basic angepaßt und beherrschen den Zeichensatz des C 64/VC 20. Andere Drucker sind mit einer Schnittstelle ausgestattet, die nicht direkt an einen Commodore-Computer anschließbar ist. Hier haben sich hauptsächlich zwei Standards gebildet. Zum einen ist das die serielle RS232C-(V.24) und zum anderen die Centronics-Schnittstelle. Drucker beider Schnittstellenarten sind trotzdem an den C 64/VC 20 anschließbar, allerdings nur wenn ein Interface verwendet wird. Dieses Interface übernimmt die Aufgabe, die vom Computer kommenden Zeichen so aufzubereiten, daß der Drucker sie richtig verarbeiten kann. Solche Interfaces können entweder in den Drucker eingebaut, oder zwischen Computer und Drucker geschaltet werden. Es ginge zu weit, alle Besonderheiten zu beschreiben. Zusammengefaßt läßt sich aber sagen, daß ein Interface mindestens über einen Linearkanal (Daten ohne Wandlung übertragen), eine wählbare CBM-ASCII-Codewandlung und eine Umschaltmöglichkeit zwischen Groß- und Kleinschreibung wie bei den Commodore-Druckern verfügen sollte (Sekundäradresse 7). Letztere Funktion ist besonders dann wichtig, wenn fertige Softwarepakete wie Dateiverwaltung und Textverarbeitung verwendet werden sollen, denn diese Programme verwenden meist den Groß-/Kleinschrift-Modus. In vielen Fällen reicht es bereits aus, zwischen Computer und Drucker ein einfaches Kabel zu legen und die Anpassung der Datenausgabe im Computer vorzunehmen. Viele Interfaces sind nach diesem Prinzip aufgebaut, man erkennt sie an der immer notwendigen Software, die zusätzlich geladen werden muß.

— Grafikfähigkeit

Ein Drucker ist immer dann grafikfähig, wenn er in der Lage ist, Daten als Bitmuster zu interpretieren. Fast alle Matrixdrucker verfügen über diese Fähigkeit. Diese Funktion ist beispielsweise dann unumgänglich, wenn Hardcopies eines Grafikbildschirmes ausgedruckt werden sollen.

— Zeichensatz

Der Zeichensatz ist verantwortlich für das Aussehen und die Zusammensetzung der Druckzeichen. Viele Drucker verfügen über internationale Zeichensätze (zum Beispiel mit deutschen Umlauten) oder über den gleichen Zeichensatz wie Commodore-Computer. Sollen beispielsweise viele Programmlistings ausgedruckt werden, ist der CBM-Zeichensatz inklusive der reversen Steuerzeichen notwendig.

— Schriftbild

Das Schriftbild ist das wesentlichste Kriterium für einen Drucker. Wichtig sind vor allem echte Unterlängen und die Punktdichte. Eine hohe Punktdichte läßt die Zeichen so erscheinen, als ob sie nicht mehr aus verschiedenen Punkten, sondern aus einer ununterbrochenen Linie bestehen. Ein weiteres Kriterium ist die Möglichkeit der Schriftbeeinflussung. Tabelle 2 gibt Aufschluß über verschiedene Methoden der Schriftgestaltung. Wichtigster Punkt ist das Vorhandensein einer Proportionalschrift, bei der die Zeichenzwischenräume so gewählt werden, daß ein harmonisches Schriftbild entsteht. Krönung der Schriftarten ist eine Near Letter Quality-Schrift.

— Geschwindigkeit

Vergleichen Sie bei diesem Punkt vor allem die Geschwindigkeit bei verschiedenen Schriftarten. Allerdings gilt: Je höher die maximale Druckgeschwindigkeit, desto schneller sind auch die Geschwindigkeit bei verschiedenen Schriftarten.

— Papierarten

Welche Papierarten kann der Drucker verwenden. Es gibt Einzelblätter (wie bei der Schreibmaschine) Rollenpapier (besonders billig), Traktor-Endlospapier (meistverwendet) und Spezialpapier (für Thermodrucker).

Fortsetzung auf Seite 160

- ☐ Grafikfähigkeit
- ☐ Zeichensätze (CBM, internationale Zeichen)
- ☐ Art der Schnittstelle (CBM, RS232, Centronics)
- ☐ Druckgeschwindigkeit: in Zeichen pro Sekunde
- ☐ Bidirektionaler Druck mit Druckwegoptimierung
- ☐ Papierarten: Thermo,- Rollen-, Traktor-, Einzelblattpapier
- ☐ Sonderfunktionen: siehe Tabelle 2
- ☐ Interface für Commodore verfügbar?
- ☐ Lebensdauer des Druckkopfes: in MTBF-Stunden = Mean Time between Failure
- ☐ Bedienungsfreundlichkeit: Drucktasten für Zeilen- und Seitenvorschub, Erreichbarkeit der DIL-Schalter zur Auswahl einiger Dauerfunktionen
- ☐ Servicefreundlichkeit
- ☐ Größe des Pufferspeichers:
- ☐ Geräuschpegel
- ☐ Bedienungsanleitung: Umfang, in deutscher Sprache
- ☐ Preis: Welche Zusatzeinrichtungen sind im Preis eingeschlossen? Zum Beispiel ein Interface

Tabelle 1. Das Leistungs- und Anforderungsprofil hilft bei der Druckerwahl

1. Schriftarten
- ☐ Korrespondenzdruck
- ☐ Fettdruck
- ☐ Doppeldruck
- ☐ Eliteschrift
- ☐ Proportionalschrift
- ☐ Picaschrift
- ☐ vergrößerte Schrift
- ☐ Unterstreichfunktion
- ☐ NLQ-Schrift (Near Letter Quality)
- ☐ Sub- und Superscript (Hoch- und Tiefstellen)
- ☐ Kursivschrift
- ☐ komprimierte Schrift
- ☐ Mischfunktion verschiedener Schriftarten
- ☐ reverser Druck
- ☐ doppelt hoher Druck

2. Sonderfunktionen
- ☐ Grafikfähigkeit mit verschiedenen Punktdichten
- ☐ Einstellen des Zeilenvorschubes
- ☐ Seitenvorschub
- ☐ Festlegen der Papierlänge
- ☐ Horizontale und vertikale Tabs
- ☐ Vorwärtsschritt um mehrere Zeilen
- ☐ Setzen des linken und rechten Randes
- ☐ Rückwärtsschritt
- ☐ ladbarer Zeichensatz
- ☐ internationaler Zeichensatz
- ☐ Papierendeerkennung
- ☐ programmierbarer Druckerreset
- ☐ Abschalten des bidirektionalen Drucks
- ☐ Rückwärtstransport des Papiers
- ☐ Reduzierung der Druckgeschwindigkeit zur Geräuschminderung

Tabelle 2. Schriftarten und Sonderfunktionen bestimmen den Ausstattungskomfort eines Druckers

Vergleich: Drucker unter 700 Mark

Lohnt sich der Kauf eines Billigdruckers? Wir haben sieben Drucker der unteren Preisklasse getestet und wollen Ihnen eine Antwort auf diese Frage geben.

Billige Drucker gibt es eigentlich nicht. Selbst einfachste Modelle bewegen sich oft auf einem Preisniveau, das über dem des C 64 liegt. Der Kauf des ersten Druckers will also wohl überlegt sein, denn mit dem falschen Drucker sind schnell einige hundert Mark verloren. Dabei ist es keineswegs gleichgültig, für welches Modell einer Preisgruppe man sich entscheidet, denn Leistungen und Schriftbild sind oft von Gerät zu Gerät stark unterschiedlich. Auch sollte man sich vor dem Kauf genauestens über die Vor- und Nachteile der verschiedenen Konstruktionsprinzipien klar werden. Wir haben für Sie einige Low-Cost-Drucker mit Preisen bis zu 700 Mark getestet.

Bunt gemischt

Zum Test standen sieben Geräte verschiedenster Bauarten, die sich in zwei Gruppen einteilen lassen. Die erste Gruppe verwendet schmales, 12 Zentimeter breites Papier. Die zweite Gruppe setzt sich aus Druckern zusammen, die mit normal breitem Papier (21 Zentimeter) arbeiten. Uns kam es beim Test vor allem darauf an, den jeweiligen Drucker in dem Anwendungsgebiet zu testen, für das er konstruiert wurde. Es ist sinnlos, einen mit schmalem Papier arbeitenden Drucker auf seine Fähigkeiten bei der Textverarbeitung hin zu testen. Niemand wird auf die Idee kommen, Briefe im Format eines besseren Kassenzettels zu verschicken. Besonders wichtig war für uns die Handhabungsfreundlichkeit und natürlich das Zusammenspiel mit dem C 64. Die Qualität des Handbuches wurde besonders berücksichtigt, da preiswerte Drucker in der Regel von Einsteigern benutzt werden.

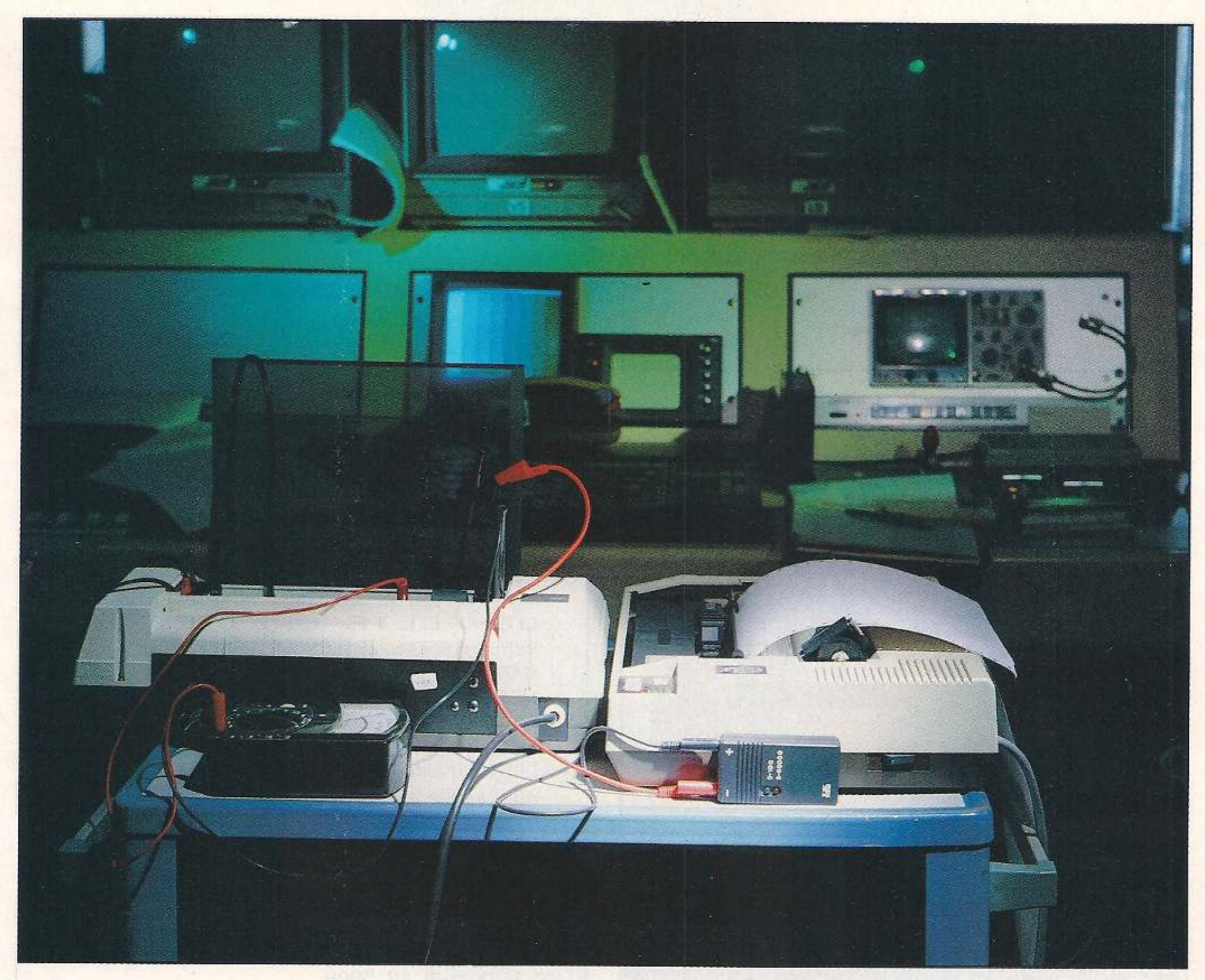
Kleiner Epson ganz groß

Was zunächst wegen seiner kompakten Maße wie ein verkleinertes Modell aussieht, erweist sich in der Praxis schnell als ein recht vielseitiges und zuverlässiges Werkzeug. Der Epson P-40 (Bild 1) ist ein Kleindrucker mit einer Papierbreite von 11,2 Zentimetern, der ursprünglich für den HX-20 Hand-Held-Computer entwickelt wurde. Er arbeitet nach dem Thermo-Prinzip und ist deswegen extrem leise. Im flachen Gehäuse präsentiert er sich wie eine verkleinerte Ausgabe des bekannten FX-80-Druckers. In der Tat teilt der P-40 mit seinem großen Bruder nicht nur die Centronics-Schnittstelle, sondern auch einige Steuerbefehle. Für einen Drucker dieser Preisklasse ungewöhnlich sind Befehle zum Einstellen der verschiedenen Schriftarten (Bild 2) wie komprimierter (80 Zeichen) und gedehnter (40 Zeichen) Schrift. Wer das gut strukturierte und umfangreiche Handbuch studiert, stößt sogar auf zwei Befehle für einfache und doppelte Grafik, die den Befehlen der »großen« Brüder entsprechen. Ganz erstaunlich ist auch der eingebaute Zeichengenerator. Er bietet die Möglichkeit zwischen verschiedenen internationalen Zeichensätzen zu wählen, unter anderem auch einem deutschen. Dieser Zeichensatz enthält 96 ASCII-Zeichen, inklusive der Groß- und Kleinschreibung. Trotz seines guten Konzeptes ist der Betrieb des P-40 am C 64 nicht ohne Probleme. In jedem Fall wird ein zusätzliches Interface notwendig, das mindestens 50 Mark kostet. Für eine riesige Auswahl solcher Schnittstellen ist allerdings gesorgt: Da der P-40 über die gleiche Befehlssyntax wie seine »großen Brüder« verfügt, können alle für die RX-80/FX-80 konstruierten Schnittstellen verwendet werden. Mittels Batterien kann er auch ohne Netzteil betrieben werden. Insgesamt ist der P-40 ein gelungenes Gerät, bei dem es allerdings am rechten Einsatzgebiet fehlt. Für eine Textverarbeitung ist sein Papier zu schmal und als Listingdrucker fehlt ihm der Commodore-Zeichensatz. Mit einem Preis von 448 Mark (ohne Interface) ist der P-40 auch etwas teuer.

Doppeltes Lottchen

Der Brother HR-5 (Bild 3) ist ein Thermo-Transfer-Drucker, den es in zwei verschiedenen Ausführungen gibt. An einem »C« hinter dem Namen erkenntlich, stellt sich die direkt an den C 64 anschließbare Version vor. Ein eingebautes Interface sorgt für alle Anpassungen, die für den Betrieb am C 64 wichtig sind. Beim HR-5 ohne »C« stehen zwei Schnittstellen, Centronics parallel oder V.24 (RS232C) zur Verfügung. Das Druckverfahren des HR-5C ist etwas ungewöhnlich. Während des Druckes fährt der Druckkopf am stillstehenden Farbband entlang und preßt es gegen das Papier. Dabei werden die angesteuerten Punkte auf dem Druckkopf erwärmt und die Farbpartikel bleiben auf dem Papier hängen. Nach dem Druck einer Zeile hält der Druckkopf an und es wird mit verhältnismäßig lautem Geräusch das Farbband weitergespult. Da der Druck bidirektional abläuft, wird die nächste Zeile in der Regel von rechts nach links gedruckt. Beim HR-5C können zwei verschiedene Papiersorten verwendet werden. Neben dem Druck auf normalem Papier mit Farbband (Bild 4) kann der Drucker auch direkt auf Thermopapier drucken. Man hat also die Wahl zwischen teurem Farbband und billigem Papier oder teurem Thermopapier. Der HR-5C ist zum Betrieb mit vier Monozellen vorgesehen. Wahlweise kann auch ein Netzgerät, das aber mit 40 Mark extra bezahlt werden muß, verwendet werden. Diese Anschaffung ist aber ratsam, denn bei einer Leistungsaufnahme von 6 Watt sind Batterien natürlich schnell erschöpft. Sehr viel Fingerfertigkeit verlangt das Einstellen der unter der Führungsstange und dem Steuerriemen verborgenen DIL-Schalter und das Einlegen der Farbbandkassette. Das eingebaute Interface wurde in wesentlichen Punkten an die Steuerung der Commodore-Drucker angepaßt. Mit der Sekundäradresse 0 erfolgt der Ausdruck im Normalmodus (Großbuchstaben und Grafikzeichen). Mit der Sekundäradresse 7 erreicht man den Zeichensatz mit großen und kleinen Buchstaben. Zu den Fähigkeiten des HR-5C gehört auch der Druck von reversen und vergrößerten Zeichen. Das umfangreiche Handbuch erleichtert die Einarbeitung in den HR-5C.

Der HR-5C ist mit einem Preis von 499 Mark sicherlich kein schlechter Kauf für alle, denen es auf problemlosen Anschluß und niedrigen Geräuschpegel ankommt. Drei Dinge sind es aber, die den sonst guten



Auf Herz und Nieren werden die Testkandidaten überprüft

Eindruck des HR-5C schmälern: Die relativ hohen Unterhaltskosten, die niedrige Druckgeschwindigkeit (10 bis 30 Zeichen pro Sekunde) und die unpraktische Handhabung.

Die Hausmarke

Seikosha und Commodore-Drucker sind zwei Worte für den gleichen Begriff. Drucker dieser Firma haben den C 64, und vorher den VC 20, auf weiten Strecken ihrer Entwicklung begleitet. Seikosha war auch die erste Firma, die außer Commodore selbst, direkt an den seriellen Port des C 64 anschließbare Drucker anbot. Kein Wunder, denn die Commodore 1525- und MPS 801-Drucker werden eigentlich von Seikosha produziert (nur das Gehäuse stammt von Commodore). So kommt es auch, daß einige unserer Testkandidaten auffällige Ähnlichkeiten besitzen. Wir haben den MPS 801 und den Seikosha GP500A miteinander verglichen. Die Mechanik beider Drucker ist identisch und auch beim Gehäuse bestehen kaum Differenzen. Der Unterschied liegt im Verborgenen, denn der MPS 801 wurde mit den gleichen Steuerbefehlen wie der seit langem bekannte 1525 (baugleich mit Seikosha GP100VC) ausgestattet. Dazu aber später mehr. Betrachten wir zunächst den GP500A (Bild 5). Mit einer Centronics-Schnittstelle ausgestattet, ist der GP500A nicht direkt an den C 64 anschließbar. Es wird deshalb notwendig, zusätzlich ein Interface anzuschaffen. Einziger Vorteil dieses Druckers gegenüber dem MPS 801 wäre der vorhandene deutsche Zeichensatz. Von dem kann der C 64-Besitzer aber wenig Gebrauch machen, denn das Schriftbild (Bild 6) ist eigentlich nicht ausreichend. Der GP500A kann keine Unterlängen drucken. Buchstaben wie »p« oder »y« werden immer angehoben, was einem harmonischen Textbild nicht gerade zuträglich ist. Wer ihn zum Programmieren verwenden möchte, stößt recht bald auf die Grenzen. Außer einer vergrößerten Schrift und einem Grafikmodus sind kaum Sonderfunktionen vorhanden. Der GP500A kostet 598 Mark.

Dem GP500A ähnlich ist der GP50A. Er ist ebenfalls ein Nadel-Matrixdrucker, bei dem die Papierbreite allerdings halbiert wurde. Auch er verfügt nur über eine Centronics-Schnittstelle. Das Haupteinsatzgebiet dieses Druckers wäre das eines preiswerten (398 Mark) Protokolldruckers beim Programmieren. Dazu fehlt ihm aber der Commodore-Zeichensatz. Da er diese Fähigkeit erst zusammen mit einem Interface erlangt, geht leider einiges vom Preisvorteil verloren. Die Handbücher zu den beiden Druckern sind ziemlich kurz gehalten und nicht auf das Commodore-Basic abgestimmt.

Die Problemlosen

Der MPS 801 (Bild 7) ist eine Weiterentwicklung des 1525 (baugleich

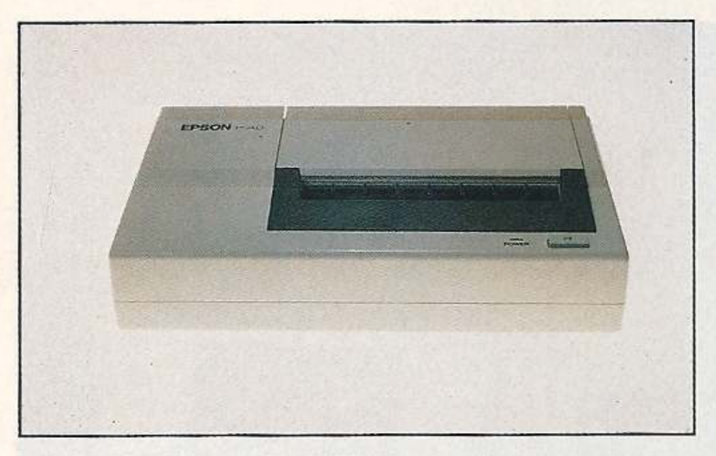


Bild 1. Epson P-40, ein Thermo-Drucker

```
EPSON P40
PAPIERBREITE: 11 ZENTIMETER
ZEICHENMATRIX: 5 X 9
DRUCKGESCHWINDIGKEIT: 40
ZEICHEN PRO SEKUNDE
GRAFIKFAEHIG: JA, ZWEI PUNKTDICHTEN
DOPPELTE BREITE
VERSCHIEDENE ZEILENABSTAENDE
KOMPRIMIERTE SCHRIFT
HERVORGEHOBENE SCHRIFT
DEUTSCH UMLAUTE:
ÄÖÜ^äöüß
```

(Schriftbild verkleinert)

Bild 2. Klein, aber mit Leistungen der Großen ausgestattet — der Epson P-40



Bild 3. Brother HR-5, ein Thermo-Transfer-Drucker

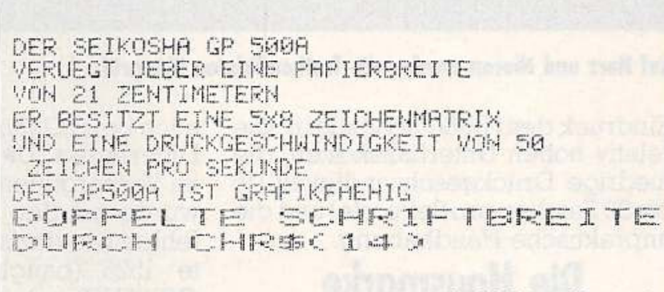
```
BROTHER HR 5 THERMOTRANSFERDRUCKER
PAPIERBREITE: 21 ZENTIMETER
ZEICHENMATRIX: 9X9
DRUCKGESCHWINDIGKEIT 30
ZEICHEN PRO SEKUNDE
GRAFIKFAEHIG: JA, ZWEI PUNKTDICHTEN
DOPPELTE BREITE
VERSCHIEDENE ZEILENABSTAENDE
KOMPRIMIERTE SCHRIFT
HERVORGEHOBENE SCHRIFT
UNTERSTRICHENE SCHRIFT
ELITE SCHRIFT MIT DEM HR-5
DEUTSCH UMLAUTE:
ÄÖÜ^äöüß
```

(Schriftbild verkleinert)

Bild 4. Gute Leistung und fast nicht zu hören — der Brother HR-5



Bild 5. Seikosha GP500A, der Nachfolger des GP100


```
DER SEIKOSHA GP 500A
VERUEGT UEBER EINE PAPIERBREITE
VON 21 ZENTIMETERN
ER BESITZT EINE 5X8 ZEICHENMATRIX
UND EINE DRUCKGESCHWINDIGKEIT VON 50
 ZEICHEN PRO SEKUNDE
DER GP500A IST GRAFIKFAEHIG
DOPPELTE SCHRIFTBREITE
DURCH CHR$( 14 )
```

(Schriftbild verkleinert)

Bild 6. Schriftprobe vom MPS 801 und GP500A



Bild 7. MPS 801 von Commodore



Bild 8. Der 1520-Plotter von Commodore

mit Seikosha GP100VC, der vom GP500C abgelöst wurde). Alle Steuerzeichen und Sekundäradressen des C 64 entsprechen denen des Druckers. Der Unterschied liegt im etwas modernisierten Gehäuse und einer anderen Druckmechanik (die des Seikosha GP500A). Das Farbband wurde gegenüber dem 1525-Drucker verkleinert und direkt auf dem Druckkopf in einer kleinen Kassette untergebracht. Durch die neue Mechanik ist der MPS 801 etwas schneller als sein Vorgänger geworden, er schafft jetzt 50 Zeichen pro Sekunde gegenüber 30 Zeichen pro Sekunde beim 1525. Leider haben diese Neuerungen ihren Preis, der MPS 801 kostet 698 Mark, bietet aber nur wenig Vorteile gegenüber dem 1525. Er eignet sich vor allem als Grafik-Drucker oder zum Listen eigener Programme. Für die Textverarbeitung gilt das gleiche wie für den Seikosha GP500A. Die Zeichendarstellung erreicht leider keine Briefqualität (Bild 6).

Der billigste für den C 64 erhältliche Drucker ist der 1520 (Bild 8). Genau genommen ist der 1520 eigentlich gar kein Drucker, sondern ein Plotter. Mit einer Papierbreite von 11,5 Zentimetern ist der 1520 in der Lage, mit seinen vier Farbminen sowohl Grafiken, als auch Programmlistings auszudrucken. Der 1520-Printer/Plotter ist im wesentlichen ein XY-Plotter, der mit kurzen Kugelschreiberminen arbeitet. Er wird durch Schrittmotoren angetrieben und ermöglicht präzises Zeichnen mit einer Auflösung von 0,2 Millimetern und einer Geschwindigkeit von 14 Zeichen pro Sekunde. Wie alle Commodore-Drucker wird auch der 1520 über Sekundäradressen gesteuert. Beim Auslisten von Programmen übersetzt der 1520 alle Grafik- und Steuerzeichen in unterstrichene Buchstaben. Nach einer kurzen Eingewöhnungszeit sind solche Programmlistings sogar deutlicher zu lesen, als die mit dem MPS 801 erstellten. Da der 1520-Plotter/Printer mittlerweile für weniger als 300 Mark erhältlich ist, lohnt seine Anschaffung auch dann noch, wenn schon ein Matrixdrucker vorhanden ist. Die Programmierung der Plotterfunktionen stellt für sich alleine betrachtet schon eine interessante Aufgabe dar. Erwägt man den 1520 als Protokolldrucker bei Meßvorgängen einzusetzen, gibt es kaum eine preiswertere Alternative. Die Handbücher sind in der für Commodore typischen Kürze gehalten, trotzdem erklären sie die wichtigsten Funktionen ausführlich genug.

Der leise Star

Der Star STX 80 (Bild 9) ist der leistungsfähigste Drucker dieses Testes. Als Thermo-Drucker konstruiert, ist er fast nicht zu hören. Der STX 80 schafft im bidirektionalen Druck bis zu 60 Zeichen pro Sekunde. Alle Buchstaben haben Unterlänge und sogar deutsche Umlaute sind vorhanden. Schade, daß der STX 80 nur mit Spezialpapier drucken kann, denn sonst wäre er der einzige auch zur Textverarbeitung einsetzbare Drucker. Sein Schriftbild erfüllt die Mindestanforderungen. Seine wahren Fähigkeiten zeigt der STX 80 wenn er mit dem Star-Interface an den C 64 angeschlossen wird. Bild 10 zeigt die umfassenden Möglichkeiten, die dem Programmierer dann zur Verfügung stehen. Die Befehle des Interfaces erlauben sogar einwandfreie Listings in Klarschrift (Steuerzeichen werden übersetzt).

Mit einem Preis von 595 Mark ohne Interface bietet der STX 80 viel für sein Geld. Er ist der ideale Drucker für alle, die gehobene Ansprüche stellen, denen aber Nadel-Matrixdrucker zu laut sind. Sein größter Nachteil sind die relativ hohen Kosten für das Spezialpapier. Das Handbuch für den Drucker und das Interface können als gelungene Produktbeschreibung bezeichnet werden.

Lohnt es sich?

Der Test hat gezeigt, daß, trotz einiger Lichtblicke, bei Low-Cost Druckern nach wie vor große Abstriche an Qualität und Leistungsfähigkeit gemacht werden müssen. Keines der getesteten Geräte erfüllt alle an einen Drucker zu stellenden Anforderungen in ausreichendem Maße. Die schwierigste Hürde, die Eignung zur Textverarbeitung, haben eigentlich alle Testkandidaten dieser Preisklasse nicht nehmen können. Entweder reicht die Qualität der Schrift für heutige Ansprüche kaum aus, oder die Papierbreite beziehungsweise Papierart behindert eine sinnvolle Anwendung. Trotzdem haben Low-Cost-Drucker ihren Markt, denn bei Preisen von 300 Mark aufwärts, sind sie die oft einzigen erschwinglichen Alternativen. Dennoch sollte jeder prüfen, ob er nicht doch zwei- oder dreihundert Mark mehr anlegen kann. Ab zirka 800 Mark gibt es heute schon Drucker, deren Schriftbild und Leistungsfähigkeit weit über denen der getesteten Geräte liegt. Der Wiederverkaufswert eines Druckers sinkt wegen der vielen mechanischen Teile schneller, als bei einem rein elektronischen Gerät wie einem Computer. Drucker sollte man lieber eine »Nummer zu groß« kaufen, denn mit steigenden Programmierfähigkeiten wachsen meist auch die Ansprüche.

(Arnd Wängler/hm)

Info:
Brother International, Im Rosengarten 14, 6368 Bad Vilbel, Tel. (061 93) 8050;
Star, Frankfurter Allee 1—3, 6236 Eschborn/Ts., Tel. (061 69) 701 80;
Commodore, Lyonerstr. 38, 6000 Frankfurt 71, (069) 66380;
Epson, Am Seestern 24, 4000 Düsseldorf, Tel. (02 11) 59521 10;
Microscan, (Seikosha-Drucker), Oberseering 31, 2000 Hamburg 60, Tel. (040) 6320030.

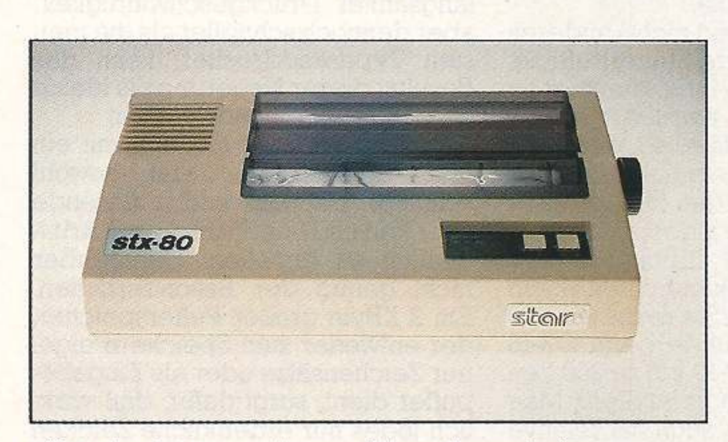


Bild 9. Star STX 80, ein leistungsfähiger Thermodrucker

```
DIE FUNKTIONEN DES STAR STX 80
DRUCKER MIT CBM INTERFACE
AUSDRUCK IN HEXADEZIMALER FORM
41 42 43 44 45 46 0D
Gross und Kleinschreibung
EIGENE GRAFIKZEICHEN

DOPPELTE BREITE
REVERSEZEICHEN

ALLE GRAFIKZEICHEN

UND HARDCOPYS MIT SIMONS-BASIC:
```

(Schriftbild verkleinert)

Bild 10. Das Schriftbild des vielseitigen und leisen Star STX 80

Superdrucker mit frechem Preis

Kaum vier Jahre, nachdem Star mit dem Gemini 10 sein Debüt auf dem Peripherie-Markt gab, wurden auf der CES in Las Vegas die Drucker einer neuen Generation vorgestellt. SG, SD und SR heißen die Modelle, die mit außergewöhnlichen Leistungen und günstigen Preisen neue Maßstäbe setzen.

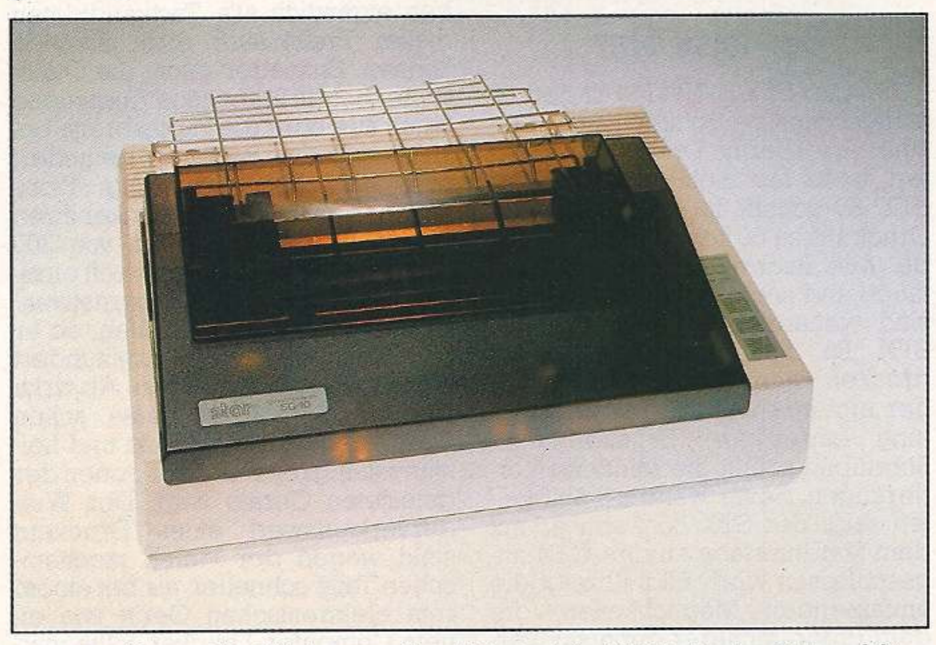


Bild 3. Der kleinste (SD-10) der neuen Star-Reihe ist keine Schönheit, aber leistungsfähig

Star setzt mit den Druckern SG-10, SD-10 und SR-10 einen neuen Standard. Die neue Star-Generation weist Leistungen auf, wie sie sonst nur von wesentlich teureren Geräten her bekannt sind. Dabei sind alle drei Geräte nach dem gleichen Konzept konstruiert. Dieses Konzept versucht, durch hohe Flexibilität, allen erdenklichen Anforderungen gerecht zu werden. Von einem modernen Drucker wird heute erwartet, daß er über verschiedene Zeichensätze verfügt, eine Reihe von Schriftarten beherrscht, voll grafikfähig ist und bei niedrigem Geräuschpegel trotzdem schnell druckt. Darüber hinaus steigen die Ansprüche an das Schriftbild. Gerade bei Briefen werden Schriftqualitäten verlangt, die mit denen eines Typenraddruckers vergleichbar sind. Alle diese Anforderungen wurden bei der Konstruktion der neuen Druckergeneration von Star berücksichtigt. Jedes Modell verfügt über einen enormen Reichtum verschiedenster Schriftarten (Bild 1) und Steuerbefehle, mit denen jeder Text abwechslungsreich aufbereitet werden kann. Alle Modelle beherrschen sechs Grafikmodi mit Auflösungen zwischen 8 x 60 und 8 x 240 Punkten pro Inch (1 Inch = 2,54 Zentimeter) und verschiedene Zeichensätze. Eine bisher wenig bekannte Neuerung ist die Definition von sogenannten »Makros«.

Diese Befehle sind nichts anderes als selbst definierte Steuerbefehle, in denen mehrere Druckfunktionen zusammengefaßt werden. So genügt es, beispielsweise eine bestimmte Schriftart einmal zu definieren, durch Aufruf des Makros steht diese Schrift dann jederzeit zur Verfügung. Auch die Druckgeschwindigkeit der Testkandidaten kann sich sehen lassen. Sie reicht von 120 Zeichen pro Sekunde beim SG-10 über 160 beim SD-10 bis zu 200 Zeichen pro Sekunde beim SR-10. Man kann diese Werte auch als relative Druckgeschwindigkeit bezeichnen, denn die tatsächliche Geschwindigkeit ist noch von einigen anderen Punkten abhängig. So verkürzen beispielsweise die Druckwegoptimierung (siehe Grundsatzartikel in dieser Ausgabe) und auch eine hohe Geschwindigkeit des Papiertransports die Zeit, nach der ein Schriftstück fertig gedruckt ist. Die drei Testkandidaten glänzen hier mit sehr guten Werten, denn sie sind alle druckwegoptimiert und transportieren das Papier mit 10 (SD/SG) bis 12 (SR) Zeilen pro Sekunde. Diese Geschwindigkeiten gelten natürlich nur für den Normalschrift-Modus, dessen Aussehen etwa dem des Epson FX-80 entspricht. Jede Veränderung der Schrift wirkt sich, wie bei allen Druckern dieses Konstruktionsprinzips, allerdings auf die Schreibgeschwindigkeit aus.

Typenraddrucker eingebaut

Bislang ließen sich mit der Proportionalschrift die besten Schriftbilder erzeugen. Bei der neuen Star-Generation wurde der ebenfalls vorhandene Proportional-Modus durch eine besondere Drucktechnik erweitert. Sie nennt sich NLQ-Schrift. Die drei Buchstaben NLQ stehen dabei für »Near Letter Quality« oder auf gut deutsch für Briefqualität oder Schönschrift. Hinter diesem Zauberwort versteckt sich ein Zeichensatz, dem eine 17 x 11-Zeichenmatrix zugrunde liegt. Der Druckkopf fährt dabei nicht nur einmal, sondern zweimal über die Zeile und setzt jedes Zeichen aus zwei Teilen zusammen. Diese entweder über einen gut erreichbaren Schalter (wie alle anderen Schalter auch) oder aber softwaregesteuert einstellbare Schrift setzt in dieser Preisklasse Maßstäbe der Druckqualität bei Matrixdruckern, wie sie bisher nur von Typenraddruckern bekannt waren (Bild 2). Zwar mit deutlich verlangsamter Druckgeschwindigkeit, aber dennoch schneller als die meisten Typenraddrucker, kann das Resultat dieser Neuerung als kleine Sensation bezeichnet werden.

Endlich wird es möglich mit ein und demselben Gerät sowohl schnelle Listings, hochauflösende Grafiken und Schriftstücke mit Briefqualität zu erstellen. Damit aber nicht genug der Besonderheiten. Ein 2 KByte großer Pufferspeicher, der entweder zum Speichern eigener Zeichensätze oder als Eingabepuffer dient, sorgt dafür, daß wirklich jedes nur erdenkliche Zeichen gedruckt werden kann. Wer will,

entwirft so seine eigenen Zeichensätze (beispielsweise andere Schriften oder wissenschaftliche Zeichen) bis hin zu ägyptischen Hyroglyphen. Als Papier kann bei allen Funktionen entweder Einzelblatt oder Endlospapier verwendet werden.

DIE FUNKTIONEN DER STAR-DRUCKER MIT CBM INTERFACE
AUSDRUCK IN HEXADEZIMALER FORM
41 42 43 44 45 46 OD
DOPPELTE BREITE
REVERSE ZEICHEN
BESONDERE EFFEKTE
64'ER MAGAZIN
Gross und Kleinschreibung
EIGENE GRAFIKZEICHEN
ALLE GRAFIKZEICHEN
UND HARDCOPYS MIT SIMONS-BASIC:

Schriftarten der neuen Star-Generation
Ein Beispiel für die Pica-Schrift
Die Elite-Schrift gehört zum Standard
Der Breitschrift-Modus
Doppeldruck ermöglicht Hervorhebungen
superscript und subskript sind für Bemerkungen gut
NLQ-Schrift im Proportional-Modus
Für viel Informationen in einer Zeile - die komprimierte Schrift

Bild 1. Die Schriftvielfalt der Star-Drucker

Mit der NLQ-Schrift (Near Letter Quality) besitzen die Star-Drucker SG-10, SD-10 und SR-10 einen bisher unerreichten Qualitätsstandard. Die NLQ-Schrift ist durchaus mit der Qualität eines Typenraddruckers zu vergleichen, allerdings bleiben die Vorteile eines Matrixdruckers ganz erhalten.

Bild 2. Mit der NLQ-Schrift wird Typenrad-Qualität erreicht

Drei (un)gleiche Brüder

Alle drei Drucker der neuen Generation sind sich sehr ähnlich. Eigentlich sind es sogar sechs Brüder, denn alle drei Geräte sind auch unter der Bezeichnung SG-15, SD-15 und SR-15 erhältlich. Sie sind dann in der Lage, Papier in einer Breite von bis zu 38 Zentimetern, gegenüber 25 Zentimetern bei den 10er-Versionen zu verarbeiten. Trotzdem sind Unterschiede auch zwischen den einzelnen Geräten vorhanden, die unterschiedliche Preise rechtfertigen. Wichtigstes Unterscheidungsmerkmal ist die Druckgeschwindigkeit. Daneben gibt es Verschiedenheiten im technischen Aufbau. Das kleinste Modell arbeitet mit Farbbandrollen, die beiden größeren verwenden Farbbandkassetten. Der Traktorantrieb ist bei den beiden kleineren Modellen anders konstruiert als beim Spitzenmodell SR-10. SD-10 und SG-10 verwenden Traktorräder, die über der Druckwalze angeordnet sind. Beim SR-10 ist der Antrieb für Endlospapier in die Druckwalze integriert. Der Vorteil dieses Konzeptes ist zum einen die flachere Bauart und zum anderen eine komfortablere Handhabung bei der Papierjustage und -entnahme. Zusätzlich wird der beim SR-10 ebenfalls vorgesehene automatische Papiereinzug möglich. Außer zwei zusätzlichen Tasten für Druckunterbrechung und Papiervorschub ist der beim SR-10 mögliche Rückwärtstransport des Papiers letzter großer Unterschied zu den kleinen Brüdern.

Anschluß sichergestellt

Die Anschlußfähigkeit an möglichst viele Computertypen ist eines der Hauptmerkmale der neuen Star-Generation. Zum Anschluß an den IBM-PC wurde sogar ein eigener IBM-Modus installiert. Die Konstrukteure haben sich weiterhin entschieden, die Drucker serienmäßig mit einer Centronics-Schnittstelle auszustatten und Interfaces für verschiedene Computer als Option (RS 232, Apple, Commodore, IEEE-488, Atari) anzubieten. Wir haben natürlich das völlig neu überarbeitete Commodore-Interface getestet. Das Interface wird am seriellen Bus des C 64 und an der Centronics-Schnittstelle des Druckers angeschlossen. Das Interface wurde so konstruiert, daß es die Vorteile aller Commodore-Drucker (Grafikfähigkeit des MPS 801, Textformatierung des MPS 802) vereinigt und zusätzlich die Fähigkeiten des Star-Druckers selbst nicht einschränkt. Sogar Hardcopy-Routinen wie sie von Supergrafik 64 und Simons Basic verwendet werden, funktionieren, ohne daß eine spezielle Maßnahme vor dem Druck ergriffen werden muß. Auch alle anderen Programme, die mit der Grafik der MPS 801-Drucker arbeiten, funktionieren einwandfrei. Trotzdem braucht der Anwender nicht auf die Vorteile, über die der Drucker selbst verfügt, zu verzichten. Der einmal eingestellte deutsche Zeichensatz bleibt, neben allen Commodore-Grafikzeichen, in vollem Umfang erhalten. So können Programmlistings und Briefe mit deutschen Umlauten hintereinander ausgedruckt werden, ohne daß auch nur ein Sonderbefehl notwendig wird.

Als positiv ist der Weg zu beurteilen, der bei der Darstellung der Commodore-Steuerzeichen in einem Programmlisting gewählt wurde. Das Interface übersetzt alle Steuer-Codes in ähnlicher Weise wie die in diesem Heft abgedruckten Programmlistings. Die Steuerzeichen werden als Klarschriftübersetzung des jeweiligen Zeichens ausgedruckt.

Die neue Referenz

Die Leistungsmarken der neuen Star-Drucker sind derzeit noch unübertroffen. Mit einem Preis von 1195 Mark dürfte der SG-10 (Bild 3) die derzeit wohl beste Empfehlung für den Commodore 64 sein. Der SD-10 mit einem Preis von 1595 Mark bietet dem Anwender, bei sonst gleicher Leistung, eine um 40 Zeichen pro Sekunde gesteigerte Druckgeschwindigkeit. Er ist deshalb auch für größere Textmengen geeignet. Der SR-10 liegt mit einem Preis von 2150 Mark sicher schon etwas außerhalb des Heim-Bereichs. Mit seinen überlegenen Fähigkeiten und der sehr hohen Druckgeschwindigkeit von 200 cps ist dieses Gerät wahrscheinlich der hauptsächlich kommerziellen Verwendung vorbehalten. Die neue Star-Generation ist zukunftsorientiert. Sie paßt sowohl zum C 64 als auch zum neuen C 128. Durch das in jeder Hinsicht gelungene Interface werden die neuen Star-Drucker sogar zu den besten »Commodore-Druckern«, die es bisher gab. Der SG-10 ist dank seiner Flexibilität und der einmaligen NLQ-Fähigkeit unsere neue Referenz in dieser Preisklasse.

(Arnd Wängler/aa)

Zu neuen Horizonten

Centronics ist nicht nur eine Schnittstelle, sondern auch ein Drucker-Hersteller. Mit dem H80A stellen wir einen Drucker der gehobenen Klasse vor.

Als die Techniker von Centronics für ihre Drucker eine parallele Schnittstelle entwarfen, konnten sie noch nicht wissen, welchen Erfolg ihre Arbeit haben würde. Mittlerweile hat sich die Centronics-Norm neben der RS 232-Schnittstelle weltweit durchgesetzt. Zum Glück wurde diese Schnittstelle von anderen Herstellern akzeptiert und ein Schnittstellenchaos verhindert. Kompromißlos haben die Konstrukteure des Horizon versucht, die Leistungen des Druckers den Ansprüchen der Kunden anzupassen. Dazu gehören natürlich mehrere Schriftarten, Grafikfähigkeit und Bedienungsfreundlichkeit. Für die Textverarbeitung spendierte Centronics dem Horizon eine spezielle Schriftart. Sie nennt sich Near-Letter-Quality-(NLQ-) Schrift und stellt das von Matrixdruckern bisher gewohnte Schriftbild in den Schatten. Die NLQ-Schrift (Schönschriftmodus) des Horizon basiert auf einer 23 x 9 Zeichenmatrix. Da der Horizon ein 9-Nadel-Matrixdrucker ist, druckt er im NLQ-Modus immer zweimal über eine Zeile. Beim zweitenmal jedoch leicht versetzt, um eine höhere Auflösung zu erreichen. Trotzdem bleibt der Horizon ein sehr schneller Drucker. Das Handbuch nennt eine Geschwindigkeit von 160 Zeichen pro Sekunde im Normalmodus. Das ist genau der gleiche Wert wie er vom Epson FX-80 erreicht wird. Wer von den beiden ist aber tatsächlich schneller? Wir haben die Probe gemacht. Beide Drucker hatten unter gleichen Bedingungen (Vizawrite 64, Kabelinterface) den gleichen Text von vier Seiten Länge auszudrucken. Das Testergebnis war überraschend: Zwei vorsorglich bereitgelegte Stoppuhren waren überflüssig, denn der Horizon schlug den Epson um Längen. Als der Horizon die letzte Zeile druckte, lag der Epson fast eine Seite zurück. Trotzdem stimmen die Geschwindigkeitsangaben in beiden Handbüchern. Der Unterschied erklärt sich aus dem schnelleren Papiertransport und der optimalen Druckkopfbewegung des Horizon.

Der Horizon ist ein vielseitiger Drucker, der die verschiedensten Papierarten verarbeiten kann. Vorgesehen sind Einzelblatteinzug und Endlospapier mit Traktorantrieb. Der Traktor ist hinter der Druckwalze versenkt und gut zugänglich. Das Konstruktionsprinzip, Papier um die Druckwalze zu schieben statt es zu ziehen, hat zwei Vorteile: Zum einen läßt sich das Papier leicht justieren, zum anderen kann es direkt oberhalb des Druckkopfes abgerissen werden. Im hinteren Teil des Druckergehäuses ist eine Vertiefung zum Einlegen einer Endlos-Papierrolle angebracht.

Wie bei fast allen Druckern können auch beim Horizon viele Druckfunktionen mit Hilfe von Schaltern eingestellt werden. Auf insgesamt drei DIL-Leisten (DIL = Dual in Line) besitzt der Horizon genau 20 Schalter. Neben den üblichen Funktionen wie Auto-Line-Feed und den internationalen Zeichensätzen, wird hier die Art der Schrift eingestellt, die beim Einschalten zur Verfügung stehen soll.

Der Horizon wird, wie die meisten Drucker, mit ESC-Befehlen und Steuerzeichen programmiert. Ein reichhaltiges Spektrum verschiedener Befehle zur Einstellung der Schriftarten, Formatsteuerungen (Tabs, linker und rechter Rand, Zeilenabstände) und für den Grafikbetrieb stehen zur Verfügung. Es sind sogar vier verschiedene Grafik-Modi mit Punktdichten von 60 bis 240 Punkten pro Inch (1 Inch = 2,54 Zentimeter) vorhanden.

Zum Laden eines eigenen Zeichensatzes mit 9 x 11 Punkte-Zeichenmatrix besitzt der Horizon einen 2 KByte großen Pufferspeicher. Da der Speicher auf insgesamt 8 KByte erweitert werden kann, besteht sogar die Möglichkeit, eigene Zeichensätze in NLQ-Qualität zu programmieren. Der Horizon verfügt über Leistungen, mit denen er allen Ansprüchen gerecht wird: Es stehen viele mischbare Schriftarten zur Verfügung. Sogar eine Proportional-Schrift, bei der alle Zeichenabstände automatisch ausgeglichen werden.

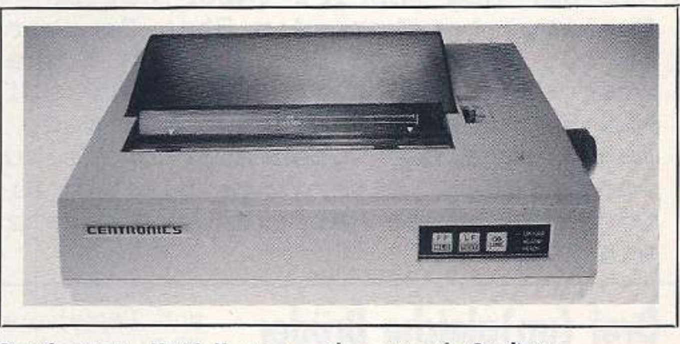
```
DER CENTRONICS H80A NLQ-DRUCKER

DIES IST DIE NORMALSCHRIFT MIT 160 CPS
DER H80A VERFUEGT AUCH UEBER NLQ-CHARACTERS

ABCDEFGHIJKLMNOPQRSTUVWXYZ!#$%&'()"
abcdefghijklmnopqrstuvwxyz1234567890

DER ITALIC-CHARACTER-SET WIRD WENIG GEBRAUCHT
MAL SEHEN WIE GUT DIE KOMPRIMIERTE SCHRIFT IST
BREIT, BREIT, BREIT
UNTERSTRICHENES WIRD DEUTLICHER
```

Eine Auswahl der verschiedenen Schriftarten des Horizon



Der Centronics H80A Horizon — überzeugende Qualität

Anschluß an den C 64

Der Horizon wurde so konstruiert, daß er mit möglichst vielen Computern zusammenarbeitet. Möchte man ihn an einen C 64 anschließen, kommt man um die Anschaffung eines Interfaces nicht herum. Will man in den Genuß der Commodore-Grafikzeichen, reverser Schrift und der Steuerzeichen in einem Programmlisting kommen, wird die Anschaffung eines Hardware-Interfaces notwendig. Ein solches Interface wird bereits von der Firma Wiesemann angeboten.

Rundum gelungen

Mit seiner hohen Druckgeschwindigkeit bei relativ niedrigem Geräuschpegel (60 dBA) ist der Horizon ein echter Profi. Der größte Nachteil des Horizon ist sein relativ hoher Preis von 2050 Mark. Dazu kommen die Kosten für ein Interface. Dafür ist der Horizon aber auch ein hochwertiges Qualitätsprodukt, das seinem Besitzer gute Dienste leistet.

(Arnd Wängler/hm)

Info: Centronics Data Computer, Lyoner Str. 44-48, 6000 Frankfurt 71, Tel. (069) 6666748

Beeindruckend: D-80X

Gute Drucker müssen nicht teuer sein. Der D-80X überrascht durch sehr gute Schriftqualität und viele Anschlußmöglichkeiten bei günstigem Preis.

Eigentlich ist der Matrixdrucker D-80X (Bild 1) ein alter Bekannter. Schon kurz nach der ersten Druckprobe war sich die Redaktion einig: »Dieses Schriftbild hatten wir doch schon mal!«. Eine nähere Untersuchung der Druckmechanik schaffte Klarheit. Sowohl der Druckkopf, als auch der Druckmechanismus sind mit dem des Commodore MPS 802 identisch. Die Ähnlichkeit ist schnell erklärt, wenn man etwas hinter die Kulissen der Elektronik-Branche schaut. Dort ist es durchaus üblich, Teile großer Zubehörlieferanten (meist in Fernost) einzukaufen, in eigene Entwicklungen einzubauen und das Ganze dann mit neuem Namen zu verkaufen. So wird beispielsweise das 1541-Laufwerk von Alps-Elektronik, die auch Alpine Autoradios fertigen, an Commodore geliefert. So ist eine gewisse Verwandtschaft von MPS 802, Decam D-80X, Mannesman Tally und Shinwa SP-80 zu erkennen. Damit soll aber nicht gesagt sein, daß es keine Unterschiede zwischen diesen Druckern gibt. Im Gegenteil, ein Mercedes mit Bosch-Zündung ist ja auch etwas anderes als ein Golf mit Bosch-Zündung. Die Druckmechanik setzt nur den Rahmen des Möglichen. Der Charakter eines Druckers wird aber erst durch die Firmware, der eingebauten Software auf EPROMs, bestimmt.

Hier liegt die Stärke des D-80X. Er wurde in Richtung größter Flexibilität konstruiert und programmiert. Dabei hatten die Entwickler ein scharfes Auge für das Marktgeschehen. Schon das Handbuch mit Beispielen in Commodore-Basic zeigt, wo der D-80X die größten Absatzchancen hat. Er ist in erster Linie für die Commodore-Computer konstruiert worden. Dies belegen seine Schnittstellen. Der D-80X ist mit insgesamt vier Schnittstellen ausgestattet, von denen drei eingebaut sind und über DIL-Schalter eingestellt werden. Je nachdem wie diese Schalter eingestellt sind, belegt der D-80X die eingebaute Centronics-Buchse mit einer anderen Schnittstellennorm. Es kann gewählt werden zwischen normaler Centronics-Schnittstelle mit ASCII oder CBM-Zeichensatz, einer IEEE-488 Parallel und einer IEC-seriellen Schnittstelle. Eine RS232 kann nachträglich eingebaut werden. Das bedeutet, daß der D-80X ohne zusätzliche Erweiterungen sowohl an die »großen« CBM-Serien, als auch an den seriellen Bus des C 64 direkt anschließbar ist. Einzige Maßnahme ist das Umlegen eines DIL-Schalters. Der Platz dieser Schalter ist allerdings eine Zumutung. Um sie zu erreichen, muß zuerst der Gehäusedeckel abgeschraubt werden. Nach längerem Suchen findet man die beiden Schalter an entlegenen Ecken der Hauptplatine. Es wäre empfehlenswert, diese Schalter gleich ab Werk an der Gehäuserückseite anzubringen.

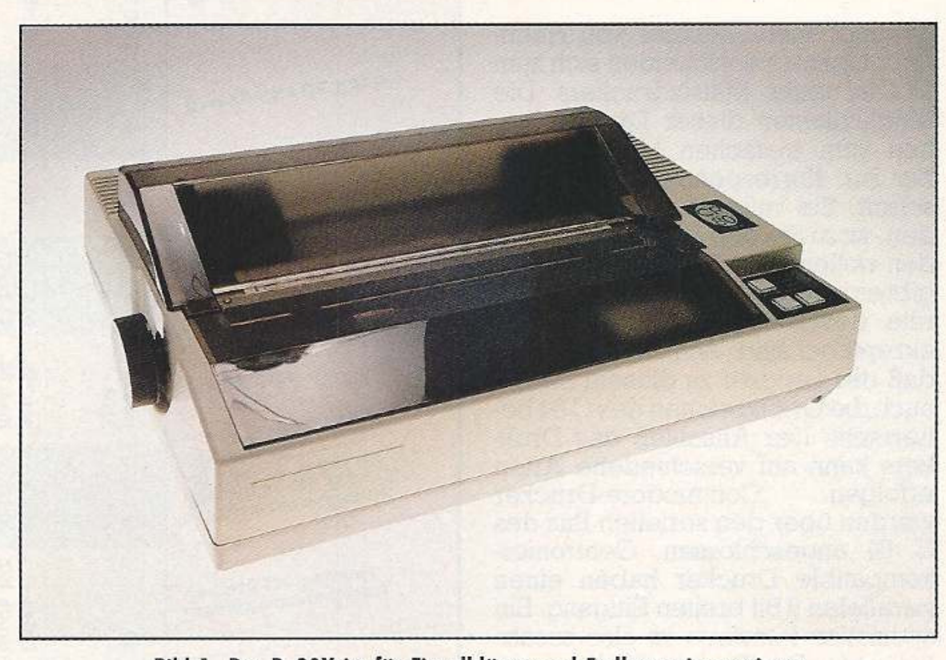


Bild 1. Der D-80X ist für Einzelblätter und Endlospapier geeignet

```
Der Matrixdrucker D 80X

Beherrscht Commodore-Grafikzeichen

Der D 80X druckt fett und
sehr breit.
Reversdruck wie bei CBM-Druckern
Die komprimierte Schrift ist gut leserlich
Unterstreichen mit CHR$(45)

Hat       eine          Tabulatorfunktion
Der Fettdruck hebt wichtiges hervor
Auch der Doppeldruck schafft Klarheit
Hochgestellt und tiefgestellt
Er beherrscht deutsche Umlaute:
Ä Ö Ü ä ö ü ß $
```

Bild 2. Die Sonderfunktionen des D-80X beinhalten das Beste der Commodore- und Epson-Drucker. (Verkleinerte Schrift)

Der Anschluß an den C 64 ist so problemlos wie bei einem CBM-Drucker; das mitgelieferte Kabel wird einfach an das Diskettenlaufwerk oder direkt am Computer angeschlossen.

Welche Leistungen kann der Programmierer von einem mit 899 Mark relativ preisgünstigen Drucker erwarten? Eignet er sich für Grafikausdruck und Textverarbeitung gleichermaßen? Beim D-80X ist es leicht, diese Fragen mit einem deutlichen »Ja« zu beantworten. Neben dem vom MPS 802 bekannt guten Schriftbild mit echten Unterlängen bietet der D-80X viele Sonderfunktionen und Schriftarten (Bild 2).

Von jedem das Beste

Die Konstrukteure haben bei der Programmierung des D-80X die Flexibilität der Hardware fortgeführt. Alle wichtigen Sonderfunktionen wie Unterstreichen, Schriftarten, Tabulatoren und Formatsteuerungen werden durch ESC-Befehle eingestellt. Die Syntax und Bedeutung der Befehle entspricht dabei im wesentlichen denen der Epson-Drucker. Zusätzlich wurden spezielle Codes zum Einstellen Commodore-spezifischer Funktionen definiert. Der reverse Druck wird beispielsweise durch den CHR$-Code 18 erreicht. Das entspricht genau dem Wert, der auch auf dem Bildschirm für reverse Zeichen sorgt. Möchte man nun eine Zeile revers ausdrucken, genügt es,

Fortsetzung auf Seite 30

Marktübersicht: Matrixdrucker

Ständig steigt das Angebot von Matrixdruckern für Heimcomputer. Ein Grund, den Druckermarkt zu durchleuchten und Ihnen eine Einkaufshilfe zu geben.

Immer mehr Besitzer von Heimcomputern entscheiden sich zum Kauf eines Matrixdruckers. Die Möglichkeiten dieser Drucker gehen vom einfachen Listingsdruck bis zur Korrespondenz in Schönschrift. Sie müssen also entscheiden, wozu Sie den Drucker verwenden wollen. Zum Druck von Listings gibt es inzwischen sehr günstige Geräte, deren Schriftbilder durchaus akzeptabel sind. Achten Sie darauf, daß der Drucker zu diesem Zweck auch die Grafikzeichen des C 64 beherrscht. Der Anschluß des Drukkers kann auf verschiedene Arten erfolgen. Commodore-Drucker werden über den seriellen Bus des C 64 angeschlossen. Centronics-kompatible Drucker haben einen parallelen 8 Bit breiten Eingang. Ein spezielles Interface ist also zusätzlich nötig. Der Preis dafür liegt zwischen 50 und 300 Mark.

Wollen Sie HiRes-Bilder des C 64 zu Papier bringen, muß der Drucker grafikfähig sein. Das heißt, seine Nadeln müssen einzeln ansteuerbar sein.

Zum Druck von langen Listings und viel Korrespondenz ist eine hohe Druckgeschwindigkeit von Vorteil. 30 Zeichen pro Sekunde, das hört sich beim Studieren von Katalogen recht schnell an, wird aber schon beim zweiten Ausdruck zur Langweilerei.

Die Geräuschentwicklung des Druckkopfes sollte nicht außer Acht gelassen werden. 70 dB(A) um Mitternacht sind für Mietwohnungen einfach zu laut.

Wie Sie sehen, Drucker ist nicht gleich Drucker. Ein hoher Qualitätsstandard muß zwar bezahlt werden, aber es gibt durchaus erschwingliche Drucker, die über eine hohe Druckqualität in den verschiedensten Schriftarten verfügen. (hm)

a) Anbieter b) Hersteller	Modell	Anzahl der Nadeln	Zeichenmatrix	Grafikfähig	Grafische Auflösung Punkte/Zoll	Anzahl Zeichensätze/ Schriftarten	Druckgeschwindigkeit Zeichen pro Sekunde	a) Papierart: endlos (1), Einzelblatt (2), Rolle (3) b) Antrieb: Walze (1), Traktor (2)	Papierbreite	a) Durchschläge inkl. Original b) max. Druckbreite in Zeichen	Lautstärke in dB (A)	Druckpuffer (in KByte)	Abmessungen (HxBxT) in cm	Schnittstellen: RS232 = 1, Centronics = 2, C 64 = 3	a) Druckerkabel im Lieferumfang enthalten b) anschlußfertig an C 64 c) Mitgeliefertes Interface	Empfohlenes Interface	Preis inkl. MwSt.
a) **Brother** b) Brother International	M1009	9	9x9	ja	480-1920	7	50	a) 1, 2, 3 b) k. A.	max. 21,08	a) 2 b) k. A.	<60	0	7x33,3x19	1, 2, 3, CP + V.24 Dual IBM, CP + V.24 Dual Epson	a) nein b) ja c) ja	—	Centronics: 749,–; Dual IBM, Dual Epson, M1009 Commodore: 799,–
	M2024L	24	k. A.	ja	k. A.	ASCII + Int. Zeich.	Datendruck 160; Pica 80, Elite 96	a) 1, 2 b) 1, 2	min. 13; max. 38	a) 4 b) k. A.	<65	k.A.	16,5x57,0x37,5	1, 2	a) nein b) nein c) ja	—	855,–
a) **Canon Rechner** b) Canon Inc., Tokyo	A-1200	9	9x7	ja	480/Zl.	1/schmal u. breit	120	a) 1, 2, 3 b) 1, 2	max. 21,6; mit Lochrand 25,4	a) 3 b) 80	<60	1 Zeile	11,0x40,0x32	2	a) ja b) nein c) ja	—	1878,–
	A-1250	9	9x9 u. 24x16	ja	936	4/schmal u. breit u. Schönschrift	140	a) 1, 2 b) 1, 2	max. 43,2	a) 3 b) 156	<60	1 Zeile	13,0x59,8x35,0	2	a) ja b) nein c) ja	—	2554,–
a) **Commodore** b) Commodore	MPS 801	7	6x7	ja	7 Punkte/ Spalte	1	50	a) 1 b) 2	min. 11,43; max. 25,4	a) 3 b) —	k. A.	0	13,6x42,5x23,5	3	a) ja b) ja c) nein	—	795,–
	MPS 802	8	8x8	bedingt	k.A.	1	52	a) 1, 2 b) 1, 2	variabel bis 25,4	a) 3 b) k. A.	k. A.	k. A.	13x42x32	3	a) ja b) ja c) nein	—	995,–
	MPS 803	7	6x7	ja	k.A.	k.A.	60	a) 2 b) 1	max. 25,4	a) 3 b) k. A.	k. A.	k. A.	7x33x19	3	a) ja b) ja c) nein	—	k.A.

a) **Centronics Data Computer** b) Centronics	GIP	8	9x9 bzw. 18x23	ja	60, 120, 240	—	50/NLQ 12	a) 1, 2, 3 b) 1, 2	min 10.16; max. 25,4	a) 3 b) 80/bei 10 cpi	<60	1 Zeile	7x19x33	—	a) ja b) ja c) ja	—	ab 854,—
	Horizon 80	9	11x9 bzw. 23x16 (bei NLQ)	ja	60 — 240	8/—	160, 27 (NLQ)	a) 1, 2, 3 b) 1, 2	min. 12,7; max. 25,4	a) 4 b) 132	<60	1 Zeile	11x40x32	2 (1 opt.)	a) ja b) nein c) Aufpreis	—	ca. 2000,—
a) **Epson Deutschland** b) Epson Japan	RX-80	9	9x9	ja	60x8 — 240x8	96 ASCII-Z. m. Unterl., 10 zus. int. Z.sätze,	100	a) 1 b) 1, 2	min. 10,16; max. 25,4	a) 2 b) 137	66	¼	10,7x37,2x30,3	Standard: 2; Optional: 1, 3 (Software-Interface EC-64)	a) ja b) nein c) Aufpreis	Softw.-Interf. EC-64 Typ 8166	1198,—
	RX-80 F/T+	9	9x9	ja	60x8 — 240x8	10/6	100	a) 1, 2, 3 b) 1, 2	min. 10,16; max. 25,4	a) 2 b) 137	63	¼	13,4x73,2x30,3	Standard: 2; Optional: 1, 3 (Software-Interface EC-64)	a) ja b) nein c) Aufpreis	Softw.-Interf. EC-64 Typ 8166	1398,—
	RX-100+	9	9x9	ja	60x8 — 240x8	10/6	100	a) 1, 2 b) 1, 2	k.A.	a) 2 b) 233	62	¼	13,4x56,8x39,3	Standard: 2; Optinal: 1, 3 (Software-Interface EC-64)	a) ja b) nein c) Aufpreis	Softw.-Interf. EC-64 Typ 8166	1798,—
	FX-80+	9	11x9	ja	60x8 — 120x9	8/6	160	a) 1, 2, 3 b) 1, 2	min. 10,16; max. 25,4	a) 2 b) 137	63	4	10x44,10x34,7	Standard: 2; Optional: 1, 3, Commodore C64	a) ja b) nein c) Aufpreis	Softw.-Interf. EC-64 Typ 8166	1848,—
	FX-100+	9	11x9	ja	60x8 — 120x9	8/6	160	a) 1, 2 b) 1, 2	k.A.	a) 2 b) 233	63	4	15,0x59,4x35,4	Standard: 2; Optional: 1, 3 (Software-Interface EC-64)	a) ja b) nein c) Aufpreis	Softw.-Interf. EC-64 Typ 8166	2398,—
	JX-80	9	11x9	ja	480x8 — 1920x8	11/7	160	a) 1, 2 b) 1, 2	min. 10,16; max. 25,4	a) 2 b) 160	63	2	11,5x44,1x34,7	Standard: 2; Optional: 1, 3 (Software-Interface EC-64)	a) ja b) nein c) Aufpreis	Softw.-Interf. EC-64 Typ 8166	2598,—
a) **Ericsson** b) Facit AB, Schweden	Facit 4510	9	9x9-15	ja	72x100	30/8	120	a) 1, 2 b) 1, 2	max. 28	a) 3 b) 80	<63	2	12,5x42,0x34,0	1, 2	a) nein b) nein c) nein	—	1995,—
	Facit 4511	9	9-18x9-17	ja	72x100	60/8	158	a) 1, 2 b) 1, 2	max. 28	a) 3 b) 80	<63	2	12,5x42,0x34,0	1, 2	a) nein b) nein c) nein	—	2485,—
a) **Fujitsu** b) Fujitsu	DPMG 9	9	18x16	ja	144	k. A.	180	a) 1, 2 b) 1, 2	min. 4; max. 30	a) 2 b) 137	69	2	11,0x41,5x31,0	2	a) nein b) nein c) nein	Data-Becker	1818,30
a) **C. ITOH Electronics GmbH** b) C. Itoh	Riteman C plus	9	9x9	ja	60, 120	4/—	105 bei 10 cpi	a) 1, 2 b) 1, 2	min. 10,4; max. 25	a) 2 b) 80	<60	2	7,3x35,6x26,8	3	a) ja b) ja c) ja	—	998,—
a) **NEC Deutschl.** b) NEC Corp. Tokio	Pinwriter P2	18	max. 21x18	ja	240x240	10	max. 180	a) 1, 2 b) 1, 2	min. 11,4; max. 25,4	a) 4 b) 80 bei 10 Pitch; 13,6 bei 17 Pitch	62	3,5 (5,6 seriell)	12,5x41x33,5	1, 2, 3	a) nein b) nein c) nein	Wiesemann, Wuppertal	2530,—

a) Anbieter b) Hersteller	Modell	Anzahl der Nadeln	Zeichenmatrix	Grafikfähig	Grafische Auflösung Punkte/Zoll	Anzahl Zeichensätze/ Schriftarten	Druckgeschwindigkeit Zeichen pro Sekunde	a) Papierart: endlos (1), Einzelblatt (2), Rolle (3) b) Antrieb: Walze (1), Traktor (2)	Papierbreite	a) Durchschläge inkl. Original b) max. Druckbreite in Zeichen	Lautstärke in dB (A)	Druckpuffer (in KByte)	Abmessungen (HxBxT) in cm	Schnittstellen: RS232 = 1, Centronics = 2, C 64 = 3	a) Druckerkabel im Lieferumfang enthalten b) anschlußfertig an C 64 c) Mitgeliefertes Interface	Empfohlenes Interface	Preis inkl. MwSt.
a) **Macrotron** b) Macrotron	Speedy 100-80	9	9x8	ja	72-144	3	100	a) 1, 2 b) 1, 2	min. 10; max. 25,5	a) 4 b) 142	58	2-4	14,0x38,0x29,5	1, 2, 3	a) ja b) ja c) ja	—	999,—
	Speedy 130-80	9	8x9	ja	81,6; 163; 330	3	130	a) 1, 2 b) 1, 2	max. DIN A4	a) 4 b) 142	58	2-4	14,0x38,0x29,5	3	a) ja b) ja c) ja	—	1099,—
	Speedy 130-136	9	9x9	ja	k.A.	3	130	a) 1, 2 b) 1, 2	max. 29,9	a) 4 b) 163	58	2-4	13,0x58,6x35,8	1, 2, 3	a) ja b) ja c) ja	—	1299,—
a) **Mannesmann Tally** b) Mannesmann Tally	MT 80 +	9	9x8	ja	640 Punkte/ Zeile	10/—	100	a) 1, 2 b) 1, 2	min. 10,16; max. 25,4	a) 4 b) 142	<60	2	12,5x38,4x31,5	1, 2	a) nein b) nein c) k. A.	—	1128,—
	MT 160/ 180	9	9x7/ 18x20	ja	64x67/ 64x133	k.A.	160	a) 1, 2, 3 b) 1, 2	min. 7,62; max. 25,4/ 40,64	a) 4 b) 264	<60	2	34,8/49,1x15,8x 24,5	1, 2	a) nein b) k. A. c) k. A.	—	2850,—/3306,—
a) **Melchers** b) Shinwa	CPA-80C	9	7x8/8x9	ja	640x8,9/ 1280x8	10/6	100	a) 1, 2, 3 b) 1, 2	min. 10; max. 25,4	a) 2 b) 142	60	0,5	12,5x38,4x31,5	3	a) ja b) ja c) ja	—	898,—
	CP-80X	9	8x8/8x9	ja	640x8/ 1280x8	8/4	80	a) 1, 2, 3 b) 1, 2	min. 10; max. 25,4	a) 3 b) 142	60	0,5	12,5x37,7x29,5	2, 3, 1 als Option	a) ja b) ja c) ja	—	948,—
a) **Microscan** b) Seikosha	GP-500 VC	7	5x7	ja	480 Punkte/ Zeile	CBM-ASCII-Code	50	a) 1 b) 2	min. 11,4; max. 25,4	a) 2 b) 80	<60	1 Zeile	11,4x44,7x31,5	3	a) ja b) ja c) ja	—	598,—
	GP-550 AVC	5, 9, 10, 18, 24	5x8-24x16	ja	960	8/16 CBM-ASCII	40-96	a) 1, 2 b) 1, 2	min. 11,4; max. 25,4	a) 2 b) 139	<60	1 Zeile	11,3x42x30,5	2, 3	a) ja b) ja c) ja	—	1198,—
	GP-700 VC	8	8x8	ja	640	CBM-ASCII-Code	38	a) 1, 2 b) 1, 2	min. 11,4; max. 25,4	a) 2 b) 80	<60	1 Zeile	11,3x45x32	3	a) ja b) ja c) ja	—	1198,—
a) **Mirwald Electronic** b) Mirwald-BMC	BX 80	9	8x9	ja	85, 170	8/18	80	a) 1, 2, 3 b) 1, 2	min. 10,16; max. 25,4	a) 4 b) 142	—	142 Bytes	12,5x37,7x29,5	1, 2, 3, IEEE488	a) ja b) ja c) Aufpreis	—	988,—
	BX 100	9	9x9	ja	60, 120, 80, 72, 90, 240	8/32	100	a) 1, 2, 3 b) 1, 2	min. 10; max. 25,4	a) 3 b) 132	<59	132 Zeich.	12,0x40,0x30,0	1, 2, 3, IEEE488	a) ja b) ja c) Aufpreis	—	1198,—
a) **Neumüller** b) Quen Data	DMP 1100	9	8x9	ja	64x64	ASCII + Deutsch	100	a) 1, 2 b) 1, 2	min. 10,1; max. 25,4	a) 3 b) 142	55	1	12,5x38,0x29,5	Standard: 2; Optional: 1, 3	a) nein b) ja c) Aufpreis	—	998,—
a) **Quen Data** b) Quen Data	DMP 1100 VC	9	8x9	ja	640x8	5	100	a) 1 b) 2	min. 10,1; max. 25,4	a) 4 b) k.A.	k. A.	k. A.	12,5x38,0x29,5	2, 3	a) ja b) ja c) ja	—	998,— bis 1040,—
a) **RFI**	RFI	9	9x9/	ja	240	6	165	a) 1, 2	min. 10,1;	a) 1+2	ca. 56	2	12x40,5x30	1, 2, 3	a) —	—	1698,—

Elektronik b) RFI	DP-165		17x17					b) 1, 2	max. 25,4	b) 132					b) ja c) Auf-preis		
b) Info-runner	Riteman II	9	9x9	ja	240	5/6	160	a) 1, 2 b) 1, 2	min. 10,4; max. 25,0	a) 1+2 b) 132	60	2	73x35,6x26,8	1, 2, 3	a) nein b) ja c) Auf-preis	—	1498,—
a) **Robotron** b) Robotron	K 6313	9	9x9	ja	480x8	9	100	a) 1, 2, 3 b) 1, 2	max. 26,5	a) 2 b) 132	58,5	2 Zeil.	13x37x28	1, 2, 3	a) nein b) ja c) ja	je nach Comp.	ca. 799,—
a) **Synelec Daten-systeme** b) Comdata	M-100	9	7x8	ja	120	11/—	100	a) 1, 2 b) 1, 2	max. 25,4	a) 3 b) 142	≤60	1 Zeile	14x38x29,5	1, 2, 3	a) nein b) nein c) Auf-preis	—	895,—
a) **Star Europe** b) Star	SD-10	9	9x11	ja	60-240	6 versch. Breiten, NLQ, pro-portional	160	a) 1, 2, 3 b) 1, 2	min. 7,5; max. 25,4	a) min. 2 b) 136	60	2	15,4x39,2x35,7	2	a) nein b) nein c) Auf-preis	Star Com-mo-dore	1595,—
	SD-15	9	9x11	ja	60-240	6 versch. Breiten, NLQ, pro-portional	160	a) 1, 2, 3 b) 1, 2	min. 7,5; max. 38	a) min. 2 b) 233	60	16	15,4x54,2x35,7	2	a) nein b) nein c) Auf-preis	Star Com-mo-dore	2100,—
	SG-10	9	9x11	ja	60-240	6 versch. Breiten, NLQ, pro-portional	120	a) 1, 2, 3 b) 1, 2	min. 7,5; max. 25,4	a) min. 2 b) 136	60	2	14,5x39,2x31,5	2	a) nein b) nein c) Auf-preis	Star Com-mo-dore	1195,—
	SG-15	9	9x11	ja	60-240	6 versch. Breiten, NLQ, pro-portional	120	a) 1, 2, 3 b) 1, 2	min. 7,5; max. 38	a) min. 2 b) 233	60	16	14,5x54,2x31,5	2	a) nein b) nein c) Auf-preis	Star Com-mo-dore	1650,—
	SR-10	9	9x11	ja	60-240	6 versch. Breiten, NLQ, pro-portional	200	a) 1, 2, 3 b) 1, 2	min. 7,5; max. 25,4	a) min. 2 b) 136	60	2	11,7x41,4x34,5	2	a) nein b) nein c) Auf-preis	Star Com-mo-dore	2150,—
	SR-15	9	9x11	ja	60-240	6 versch. Breiten, NLQ, pro-portional	200	a) 1, 2, 3 b) 1, 2	min. 7,5; max. 38	a) min. 2 b) 233	60	16	11,7x55,6x34,5	2	a) nein b) nein c) Auf-preis	Star Com-mo-dore	2650,—
a) **Techni-tron** b) Mitsui	Mitsui MC2100	9	9x7	ja	k.A.	—/4	120	a) 1, 2, 3 b) 1, 2	min. 10,16; max. 25,4	a) 4 b) 80 (10 cpi)	60	2	13,6x41,8x30,4	1, 2	a) nein b) nein c) Auf-preis	Techni-tron	1705,—
	Mitsui MC2200	9	9x9	ja	9xn	9 nat. Zeich.-Sätze	180	a) 1, 2, 3 b) 1, 2	min. 10,16; max. 25,4	a) 3 b) 80 (10 cpi)	60	2	13,6x41,8x30,4	Standard: 2; Optional: 1	a) nein b) nein c) Auf-preis	Techni-tron	ab 1881,—
a) **Terhechte-Abels** b) Info-runner	Riteman C+	9	9x9	ja	k.A.	k.A.	105	a) 1, 2 b) 1, 2	k.A.	a) 4 b) k.A.	k. A.	k.A.	k. A.	3	a) ja b) ja c) —	—	998,—
	Riteman F+	9	9x9	ja	k. A.	+ NLQ	105	a) 1, 2 b) 1, 2	k.A.	a) 4 b) 80	k. A.	2 od. 8	k. A.	Standard: 2; Optional: 1, 3	a) nein b) ja c) Auf-preis	Zett C64 I/F	1148,—
	Riteman II	9	9x9	ja	k. A.	+ NLQ	160	a) 1, 2 b) 1, 2	max. 25,4	a) 4 b) 80	k. A.	2 od. 8	k. A.	Standard: 2; Optional: 1, 3	a) nein b) ja c) Auf-preis	Zett C64 I/F	1498,—
a) **Unitronic** b) Robotron	K6313	9	9x7	ja	100	8/5	100	a) 1, 2, 3 b) 1, 2	min. 8,5; max. 21,6	a) 3 b) 132	60	1 Zeile	13x28x37	1, 2	a) nein b) ja c) ja	—	890,—

Fortsetzung von Seite 25

innerhalb der Anführungsstriche einer PRINT-Zeile auf die Tasten CTRL und 9 zu drücken. Einfacher geht es wirklich nicht mehr. Auf ähnliche Weise werden Doppeldruck, Fettschrift, komprimierte Schrift, Groß- und Kleinschrift sowie Sub- und Superscript eingestellt.

Volle Grafikfähigkeit

Im Gegensatz zum MPS 802 ist der D-80X voll grafikfähig. Er besitzt sogar zwei verschiedene Punktdichten. Im ESC K-Modus arbeitet dieser 9-Nadel-Matrixdrucker mit 640 Punkten pro 7,5 Zeilen, beim ESC L-Modus sind es sogar 1 280 Punkte pro 7,5 Zeilen. Ein lange gehegter Wunsch vieler Programmierer wird erfüllt, ja sogar übertroffen: Der D-80X besitzt auch im Commodore-Zeichensatz die deutschen Umlaute. Dafür mußte zwar auf einige Grafikzeichen verzichtet werden, der Verlust fällt aber normalerweise kaum auf. Leider entsprechen die Stellen, an denen die Umlaute einprogrammiert wurden, nicht denen der Standard-ASCII-Tabelle. Es kann deshalb leicht zu Problemen mit verschiedenen Textverarbeitungsprogrammen kommen, die keine Code-Zuweisung erlauben.

Hier wäre es vorteilhaft, wenn man nicht nur bei der Centronics-Schnittstelle, sondern auch bei der seriellen IEC-Schnittstelle, zwischen ASCII- und Commodore-Zeichensatz, wählen könnte. Wünschenswert wäre auch ein Linearkanal, bei dem alle Daten an den Drucker ohne irgend eine Umwandlung übermittelt werden.

Vorbildlich und konkurrenzlos

Trotz einiger kleiner Schwächen ist der Decam D-80X der zur Zeit interessanteste, direkt an den C 64 anschließbare Drucker unter 900 Mark. Für den erstaunlich niedrigen Preis erhält man einen vielseitigen Drucker, der sowohl zum Listingausdruck als auch zur Grafik- und Textverarbeitung geeignet ist. Sein ausführliches Handbuch erleichtert auch dem weniger erfahrenen Programmierer das Kennenlernen dieses vielseitigen Druckers. Die vielen eingebauten Schnittstellen rüsten den D-80X schon heute bestens für kommende Computergenerationen. Der D-80X ist im wahrsten Sinne des Wortes »beeindruckend«. (Arnd Wängler/hm)

Info: Decam GmbH, Postfach 1232, 7505 Ettlingen, Tel. (07243) 69264, Preis 899 Mark.

Der MPS 802 lernt deutsch

Der MPS 802 ist als zuverlässiger Drucker mit ansprechendem Schriftbild bekannt. Leider kennt er keine deutschen Umlaute. Unser kleines Programm ändert das: Der MPS 802 lernt deutsch.

Die Geschichte des MPS 802 ist interessant und abwechslungsreich wie kaum eine andere. Entstanden ist er aus dem CBM 4022/4023 der für die PET-Generation geschaffen wurde und noch einen parallelen IEC-Bus (IEEE 488) besaß. Daraus entstand der 1526 (mit serieller Schnittstelle), der mit immer neuen Gerüchten über seine Fähigkeiten und seine Fehler überraschte. Von einem zweiten Modus war die Rede, und sogar von einer vollen Grafikfähigkeit. Wie aber jeder weiß, kann sowohl der 1526 als auch der MPS 802 Grafiken nur im »Zitter-Rumpel-Verfahren« erzeugen. Soll heißen, seine Grafikfähigkeit beschränkt sich auf ein einziges Zeichen, das ständig umprogrammiert wird. Seit der Hannover-Messe 1984 nennt sich der 1526 nun MPS 802 (Bild 1), ist aber rein technisch gesehen unverändert geblieben. Was neu ist, ist die sogenannte Firmware, oder um es gleich beim richtigen Namen zu nennen, das Betriebssystem. Es ist müßig, die vielen Fehler der verschiedenen 1526-Versionen aufzuzählen, beim MPS 802 sind sie jedenfalls ausgemerzt. Deshalb ist es auch für jeden Besitzer des 1526 empfehlenswert, sein Kernal-ROM durch das des MPS 802 zu ersetzen, es funktioniert einwandfrei. Der Austausch ist ein einfaches Unterfangen, aber dazu später mehr. Schauen wir uns zunächst einmal die Hardware des MPS 802 an. Gesteuert wird das kleine Druckwunder durch einen zum 6510 softwarekompatiblen 6504-Prozessor. Außer dem 6504 befinden sich noch zwei 6532-RIOT (RAM Input/Output Timer) und eine 6522-VIA (Variable Interface Adapter) auf der Hauptplatine. Das Betriebssystem befindet sich in einem 8-KByte-EPROM vom Typ 2764, wie er überall im Handel erhältlich ist. Im einzelnen ist die Speicheraufteilung des MPS 802 in Bild 2 dargestellt.

Genug der Theorie. Im praktischen Betrieb fällt bald schon das



Bild 1. MPS 802 als Nachfolger des 1526

Fehlen der deutschen Umlaute schmerzlich auf. Dem kann abgeholfen werden. Nachstehend abgedrucktes Programm (Listing 1) verändert in der schon vom Hypra-Perfekt (Ausgabe 4/85) bekannten Overlay-Methode das Original-Betriebssystem. Dazu müssen wir allerdings erst einmal in den Besitz des Kernals kommen. Lösen Sie die vier Schrauben des Gehäuses und nehmen Sie (nach dem Farbband) vorsichtig das Gehäuseoberteil ab (Vorsicht! Garantieverlust bei neuen Geräten). Nachdem Sie sich gemerkt haben, wie das Verbindungskabel vom Deckel zur Hauptplatine eingesteckt war, können Sie auch dieses entfernen. Auf der linken, hinteren Seite des Druckers ist eine ebenfalls abschraubbare Verkleidung aus Blech angebracht — weg damit. Jetzt braucht nur noch der

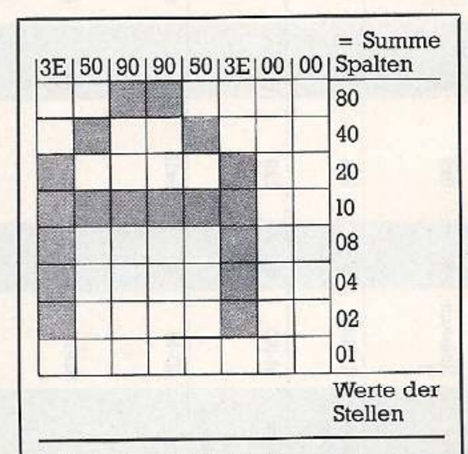


Bild 4. So wird die Zeichenmatrix des Buchstabens »A« berechnet

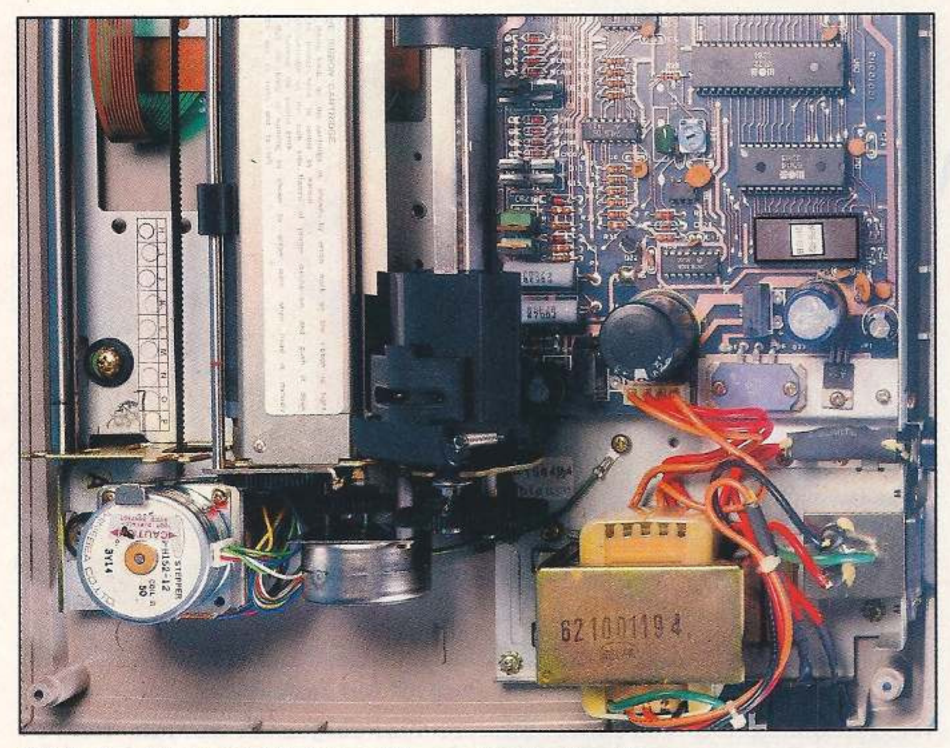


Bild 3. Der EPROM 2764 (mit dem silbernen Plättchen) muß herausgenommen und neu programmiert werden.

2764-EPROM in der rechten hinteren Ecke der Platine (Bild 3) geortet und vorsichtig herausgehebelt werden. Die Besitzer eines EPROM-Programmiergerätes wissen jetzt sicherlich schon was kommt — das EPROM wird ausgelesen und abgespeichert. Im folgenden wird deshalb davon ausgegangen, daß der Inhalt des MPS 802-EPROMs von $6000 bis $7FFF im Speicher des C 64 steht.

Die Zeichenmatrizen, daß heißt die Informationen über das Aussehen der einzelnen Zeichen, stehen nun im Bereich von $6400 bis $69FF. Aber in welcher Form? Da der MPS 802 einen Druckkopf mit 8 Nadeln besitzt, besteht jedes Zeichen aus 8 mal 8 Punkten. Jedes Zeichen belegt also 8 Byte. Jedes Byte definiert eine Spalte des Zeichens, da die Nadeln im Druckkopf senkrecht angebracht sind. Ist ein Bit gesetzt, so wird die entsprechende Nadel beim Druck des Zeichens aktiviert. Steht das entsprechende Bit auf Null, wird natürlich auch kein Punkt gedruckt. Das Leerzeichen besteht beispielsweise aus acht mal Byte 0. Etwas deutlicher wird das Ganze, wenn wir uns einmal das Zeichen »A« genauer anschauen. Die Zeichenmatrix für das A steht von $6408 bis $640F.

Ein Monitor liefert uns für diesen Bereich die Werte:
3E 50 90 90 50 3E 00 00

Jede Hexadezimalzahl repräsentiert die Summe einer Spalte (Bild 4). Man erkennt, welcher Zusammenhang zwischen der Hexadezimalzahl (Bitmuster) und der gedruckten Matrix besteht. Das Programm aus Listing 1 verändert die vorhandene Zeichenmatrix so, daß es die deutschen Umlaute an Stelle einiger Grafikzeichen zusammen mit der deutschen Version von Vizawrite druckt. Außerdem werden die Steuercodes für Breitschrift von CHR$(1) auf CHR$(14) und die Rückstellung auf Normalschrift von CHR$(129) auf CHR$(16) geändert, was gebräuchlicher ist. Wer aber dennoch selber den Zeichensatz abändern möchte, findet die einzelnen Zeichensätze an folgenden Speicherstellen:

```
10 POKE 56,96:POKE 55,0:CLR                   <020>
20 PRINT"{CLR,3DOWN,4SPACE}MPS MIT DEUTSCH
   EM ZEICHENSATZ"                            <123>
30 PRINT:PRINT"ORIGINAL MPS-KERNAL VORHER
   IM SPEICHER"                               <188>
40 PRINT"VON $6000-$7FFF EINLADEN UND NEW{
   8SPACE}EINGEBEN!!!!"                       <014>
50 PRINT                                      <203>
80 PRINT"LESEN DER DATA-ZEILEN":PRINT:PRIN
   T                                          <051>
100 T=0                                       <154>
110 T=T+1:READ A:IF A=0 THEN 240              <202>
120 READ B: REM ANZAHL DER BYTES              <042>
130 READ P1: REM PRUEFSUMME                   <092>
140 P2=0:PRINT"BLOCK ";T;"{2SPACE}";          <187>
150 FOR I=A-OF TO A-OF-1+B                    <128>
160 READ D:POKE I,D:P2=P2+D                   <211>
170 NEXT I                                    <117>
180 IF P2<>P1 THEN 210                        <224>
190 PRINT"{3SPACE}OK"                         <053>
200 GOTO 110                                  <227>
210 PRINT"PRUEFSUMME FALSCH: ";P2;" STATT
    ";P1:PRINT                                <254>
220 GET A$:IF A$=""THEN 220                   <061>
230 GOTO 110                                  <001>
240 PRINT:PRINT"FERTIG!":END                  <060>
300 REM *** AB HIER DATAS ***                 <208>
305 REM *****************                     <139>
310 REM **** BLOCK 1 ****                     <178>
315 REM *****************                     <149>
320 DATA 25600,8,609                          <240>
321 DATA 0,88,165,165,165,26,0,0              <053>
325 REM *****************                     <159>
330 REM **** BLOCK 2 ****                     <199>
335 REM *****************                     <169>
340 DATA 25816,24,1852                        <108>
341 DATA 190,80,144,144,80,190,0,0,188,66,
    66,66,66,188,0,0,188,2,2,2,2,188          <103>
342 DATA 0,0                                  <102>
349 REM *****************                     <183>
350 REM **** BLOCK 3 ****                     <220>
355 REM *****************                     <189>
360 DATA 26840,32,1889                        <135>
361 DATA 4,170,42,170,28,2,0,0,28,162,34,1
    62,28,0,0,0,60,130,2,132,62,0,0           <002>
362 DATA 0,63,64,146,146,146,108,0,0          <027>
369 REM *****************                     <203>
370 REM **** BLOCK 4 ****                     <241>
375 REM *****************                     <209>
380 DATA 32763,1,168                          <045>
390 DATA 168                                  <169>
1000 DATA 0                                   <158>
```

Listing 1. Umrüsten des MPS 802 auf den deutschen Zeichensatz

```
10 POKE 56,96:POKE 55,0:CLR                   <020>
20 PRINT"MPS-ROM IN $6000-$7FFF"              <215>
30 FOR I=12*4096 TO I+26:READ A:
   POKE I,A:NEXT I                            <009>
40 SYS 12*4096                                <168>
60 B=PEEK(32763):C=PEEK(253)                  <036>
70 D=228-C:B=B+D:IF B<0 THEN B=B
   +256                                       <050>
75 IF B>255 THEN B=B-255                      <137>
80 POKE 32763,B:PRINT"CHECKSUM A
   NGEPASST!":END                             <069>
1000 DATA 169,0,160,128,133,251,
     132,252,168,162,28,24,198,2
     52,113,251,200,208                       <086>
1010 DATA 251,202,208,246,105,0,
     133,253,96                               <160>
```

Listing 2. Anpassung der Prüfsummen

$0000—$00FF	= Zeropage
$0100—$01FF	= Prozessorstack
$0200—$03FF	= I/O-Bausteine
$0400—$1FFF	= Betriebssystem

Bild 2. Die Speicherbelegung des MPS 802

Von $6400—$6407 steht der Klammeraffe. Darauf folgen die Großbuchstaben von A—Z ($6408—$34DF). Ab $34E0 kommen die Sonder- und Grafikzeichen. Die Kleinbuchstaben beginnen bei $6808 (mit dem kleinen a). Leider hat der Drucker kein RAM, mit dem man eine Veränderung des Zeichensatzes ausprobieren könnte. Beim Entwerfen des eigenen Zeichensatzes ist ferner auf folgendes zu achten: Der Drucker führt nach dem Einschalten einen Selbsttest durch. Dabei wird auch das ROM geprüft, indem die Prüfsumme über den Bereich $6400 bis $7FFF ermittelt wird. Listing 2 dient in diesem Fall dazu, die Prüfsumme eines bereits veränderten, im Bereich von $6000 bis $7FFF stehenden ROMs anzupassen. Diese Anpassung ist natürlich nur dann notwendig, wenn eigene Veränderungen vorgenommen wurden.

Die noch verbleibenden Arbeiten sind schnell durchgeführt. Mit einem EPROM-Programmiergerät wird ein 2764-EPROM mit dem Inhalt der Speicherstellen $6000 bis $7000 gebrannt. Das neue Betriebssystem wird dann anstelle des alten in den Drucker eingesteckt (Kerbe auf Kerbe). Ob alles programmgemäß abgelaufen ist, läßt sich am einfachsten mit dem Selbsttest feststellen. Sollten Sie nun unter den vielen Zeichen tatsächlich die deutschen Umlaute finden, dann dürfen Sie sich ruhig ein kleines Päuschen genehmigen, denn Sie haben nun etwas Einzigartiges: Den MPS 802 in deutscher Version.

(Ernst Schöberl/Arnd Wängler/gk)



DIE VIDEOWERKSTATT



Heimvideo mit dem C 64. Leichte Nachbearbeitung mit dem Koala-Painter.

Mit Computer, Kamera und Video-Digitizer erschließen sich neue Dimensionen — der C 64 lernt das »Sehen«. Ungeahnte Möglichkeiten stecken hinter dieser Technik.

Der Video-Digitizer von Köhler (Österreich) ist ein kleines, in ein Kunststoffgehäuse eingebautes Modul, das am User-Port des C 64 angeschlossen wird. Von dort erhält es auch seine Stromversorgung. Einziger zusätzlicher Anschluß ist eine Cinch-Buchse auf der Oberseite des Moduls. Mit einem entsprechenden Kabel wird hier ein Norm-Videosignal (zirka 1 V_{SS} an 75 Ohm, BAS/FBAS-Signal) eingespeist.

An dieser Buchse lassen sich Geräte wie beispielsweise ein Videorecorder oder eine Videokamera (eventuell unter Zwischenschaltung des Recorders) anschließen. Auch der Videoausgang eines Fernsehgerätes ist geeignet. Wir haben das VD 64 mit einer Videokamera und einem Videorecorder getestet.

Videorecorder anschließbar

Der C 64 wird durch ein kleines Steuerprogramm von der mitgelieferten Diskette auf den Empfang der Bildinformationen vorbereitet. Dieses Programm sorgt gleich nach dem Start für die automatische Einpegelung auf das ankommende Signal. Schon kurz nach diesem Meßvorgang steht das erste Bild auf dem Monitor. Das VD 64 arbeitet extrem schnell und braucht deshalb zur Bildabtastung kein Standbild. Das lästige Anhalten des Videorecorders oder die Montage der Videokamera auf einem Stativ kann deshalb ganz entfallen. Das Bild wird in einem Durchlauf (zirka 20 Millisekunden) abgetastet und in den Multicolor-Modus des C 64 übertragen. Dieser Vorgang dauert nur etwa 0,4 Sekunden. Damit ergibt sich für den Betrachter der Grafik eine Wiederholungsrate von zwei Bildern pro Sekunde. Die Bildabtastung läuft dann so lange kontinuierlich ab, bis die Signalübertragung mit der STOP-Taste »eingefroren« wird. Das Bild kann nun durch einige interessante Befehle modifiziert werden. So verändern die Funktionstasten die Farbsättigung der gerade ausgewählten vier Farben. Im Differenzmodus wird der jeweilige Bildinhalt vom vorhergegangenen Bild subtrahiert. Dadurch treten bewegte Objekte besonders klar hervor. Sogar eine komplette Invertierung (jeder helle Punkt wird dunkel und umgekehrt) des Bildes gehört zu den Fähigkeiten des VD 64-Steuerprogramms.

Video in Multicolor

Die Konstrukteure des VD 64 haben den Multicolor-Modus des C 64 trotz seiner geringeren Auflösung (200 x 160 Punkte) gegenüber dem High-Resolution-Modus (200 x 320 Punkte) gewählt. Dafür stehen entweder vier verschiedene Graustufen, oder ersatzweise, vier Farben zur Verfügung. Durch geschickte Auswahl der Farben entstehen beeindruckende Effekte. Manche Bilder sehen wie vom Computer selbst entworfen aus, andere gleichen dem Original in verblüffender Weise. Damit diese Video-Kunstwerke aber auch einen dauerhaften Charakter bekommen, stehen sowohl eine Lade-/Speicheroption, als auch eine Hardcopy-Routine zur Verfügung. Die Hardcopy-Routine läßt die Einstellung verschiedener Druckertypen zu. Bei der getesteten Version konnten per Auswahlmenü die Drucker MPS 802, MPS 801, Hewlett Packard Ink Jet und der Commodore 4023 eingestellt werden. Weitere Anpassungen sind nach Angaben des Herstellers bereits in Arbeit.

Kompatibel zu Koala-Bildern

Der Clou des VD 64 ist aber das Format, in dem er seine Bilder ablegt. Es entspricht genau dem des Koala-Painter. Hier eröffnet sich eine unerschöpfliche Palette von Bearbeitungsfunktionen. Die Bilder können nachträglich mit dem Koala-Painter editiert oder farblich verändert werden; mit allen 16 Farben. Auch das Mischen verschiedener Bildbestandteile erweitert das Anwendungsspektrum um ein Vielfaches. Auf diese Weise können Sie beispielsweise beliebig viele »Zwillinge« ihres Motivs herstellen oder dem ganzen Bild einen neuen Hintergrund geben. Die Bilder können selbstverständlich auch in eigene Programme eingebaut oder mit einem zum VD 64 dazugehörigen Programm in der Art einer Diavorführung gezeigt werden.

Der VD 64 ist mehr als nur eine sinnvolle Ergänzung des C 64. Es ist ein Vorbote einer zukünftigen Entwicklung, in der fast alle elektronischen Medien vermehrt miteinander verknüpft werden. Es ist durchaus vorstellbar, daß die mit dem VD 64 aufgenommenen Bilder über das Telefonnetz übertragen werden. Wie vielfältig die Anwendungsgebiete des VD 64 sind, läßt sich auf den ersten Blick kaum abschätzen. Wie man den VD 64 letztendlich einsetzt und welches Bild-Motiv gewählt wird, ist reizvolle Aufgabe des Anwenders.

(Arnd Wängler/hm)

Bezugsquelle: Merkens, Fuchsstraße 6A, 6231 Schwalbach, Tel. (06196) 83030 + 3026, Preis 575,— + Mehrwertsteuer


64ER ONLINE



Mit 5 Mark zu neuen Dimensionen

Völlig neue Klangdimensionen schafft der C 64 an einer Stereoanlage. Die neue Spielegeneration mit ihrer hervorragenden Tonuntermalung gewinnt stark an Reiz. Zusätzlich können Sie nun Ihre Kompositionen mit bestmöglicher Qualität aufzeichnen.

Mit etwa 5 Mark läßt sich ein Anschlußkabel bauen, mit dem Sie den C 64 an Ihre HiFi-Anlage anschließen können. Mit 130 Watt auf jeder Seite wird nicht nur Ghostbusters zu einem neuen Erlebnis. Denn Lautsprecher von Fernseher und Monitor (falls überhaupt einer eingebaut ist) haben in der Regel nur eine bescheidene Tonqualität.

Für den geplanten Anschluß wird der Audio/Video-Port des C 64 und die DIN-Buchse der Stereoanlage oder des Verstärkers hergenommen. Sollte Ihr Verstärker nur Chinch-Buchsen haben, müssen Sie die Pins 3 und 5 des Audio-Steckers einfach durch zwei Chinch-Stecker ersetzen. Die Beschaltung der Video/Audio-Buchse des C 64 zeigt Bild 1. Entscheidend sind die Pins 2 (Masse) und 3 (Tonausgang). Sollten Sie einen älteren C 64 mit 5poligen Video/Audio-Anschluß haben, ist das kein Grund, das Handtuch zu werfen. Die Belegung der Pins 1 bis 5 beider Buchsen stimmt nämlich völlig überein. Die Beschaltung der DIN-Buchse Ihrer Stereoanlage sehen Sie ebenfalls in Bild 1. Die wichtigen Anschlüsse sind die Pins 3 (Wiedergabe links), 5 (Wiedergabe rechts) und 2 (Masse). Damit der Monoausgang des C 64 beide Stereo-Kanäle ansteuern kann, wird das Tonsignal des C 64 gleichzeitig auf Pin 3 und 5 des Stereosteckers gelegt.

Aber nun genug der Theorie. Legen Sie den Lötkolben schon mal bereit. Als Bauteile für das Adapterkabel in Bild 2 brauchen Sie zwei 5polige Diodenstecker, bei denen die fünf Pins einen Halbkreis bilden und ein abgeschirmtes 2adriges Kabel.

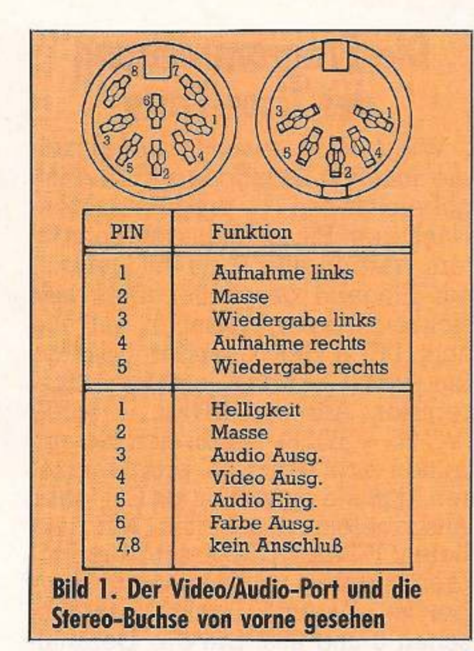
Haben Sie einen Monitor am C 64 angeschlossen, müssen Sie einen Verteiler wie in Bild 3 bauen, um die Video/Audio-Buchse des C 64 doppelt nutzen zu können. Zu diesem Zweck sollten Sie folgende Teile besorgen:

8polige Klein-Geräte-Kupplung
5poliger Diodenstecker
8poliger Klein-Geräte-Stecker
2- und 5adriges abgeschirmtes Kabel.

Die Klein-Geräte-Kupplung sieht von vorne aus wie die Video-/Audio-Buchse des C 64. Hat Ihr C 64 noch die alte 5polige Buchse, sind alle 8poligen Bauteile durch 5polige zu ersetzen.

Haben Sie alle Teile vor sich auf dem Tisch liegen, kann's losgehen. Schneiden Sie das Kabel in die gewünschte Länge und isolieren Sie die einzelnen Leitungen ab. Die äußere Plastikhülle wird am einfachsten mit einem scharfen Messer entfernt. Schneiden Sie dazu etwa 3 cm vor Kabelende die Isolation ringsum vorsichtig ein, bis Sie auf die Abschirmung stoßen und ziehen unter Drehen die Isolation ab. Jetzt wird die Abschirmung aufgetrennt, verdrillt und die einzelnen Leitungen etwa 2 mm weit abisoliert.

Bevor Sie sich nun ans Löten machen, müssen noch die Gehäuse der Stecker auf das Kabel geschoben werden.



PIN	Funktion
1	Aufnahme links
2	Masse
3	Wiedergabe links
4	Aufnahme rechts
5	Wiedergabe rechts
1	Helligkeit
2	Masse
3	Audio Ausg.
4	Video Ausg.
5	Audio Eing.
6	Farbe Ausg.
7,8	kein Anschluß

Bild 1. Der Video/Audio-Port und die Stereo-Buchse von vorne gesehen

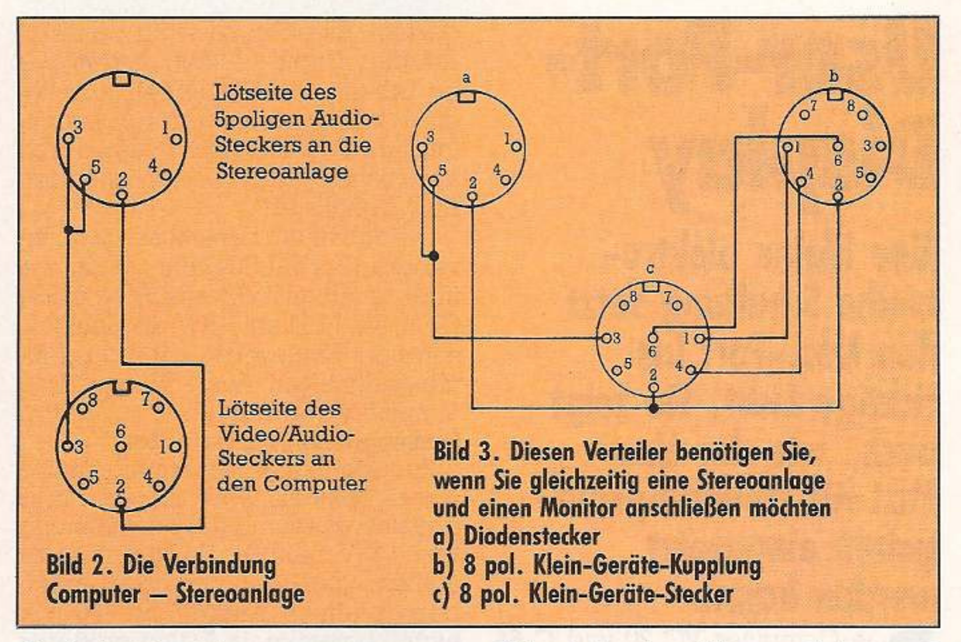


Bild 2. Die Verbindung Computer – Stereoanlage

Bild 3. Diesen Verteiler benötigen Sie, wenn Sie gleichzeitig eine Stereoanlage und einen Monitor anschließen möchten
a) Diodenstecker
b) 8 pol. Klein-Geräte-Kupplung
c) 8 pol. Klein-Geräte-Stecker

Nun können Sie die Drähte wie in Bild 2 oder 3 beschrieben anlöten. In den Schaltplänen sehen Sie die Anschlüsse von der Lötseite, wie das allgemein üblich ist. Pin 5 der Video/Audio-Buchse des C 64 sollte man mit Vorsicht genießen. Es ist der Toneingang des C 64 und führt direkt an den Sound-Chip. Wird hier unvorsichtigerweise eine Spannung angelegt, kann sich der SID schnell verabschieden. Die Verbindung zwischen Pin 3 und 5 des Audio-Steckers wird am einfachsten mit einer Lötbrücke hergestellt. Der Draht wird dazu etwas länger abisoliert und an beide Pins angelötet. Da die Kabelabschirmung nur computerseitig ans Steckergehäuse angelötet werden soll, wurde in den Schaltplänen darauf verzichtet.

Das U-förmige Blech am Steckerende dient zur Zugentlastung der Lötstellen, um ein Ausreißen zu verhindern. Das Kabel wird dazu nach dem Anlöten einfach mit einer kleinen Zange in dieses Blech-U eingequetscht und dadurch fest mit dem Stecker verbunden.

Anschluß an Chinch-Buchse

Hat Ihr Verstärker keinen DIN-Eingang, sondern Chinch-Buchsen, dann löten Sie einfach anstelle von Pin 3 und 5 im Audiostecker zwei Chinch-Stecker (Eingang linker und rechter Kanal) an, indem Sie das Kabel von einem Stecker zum zweiten weiterführen. Masse liegt bei einem Chinch-Stecker immer auf dem äußeren Kontakt.

Nach einer Schlußüberprüfung des Adapterkabels steht der neuen Klangdimension nichts mehr im Wege. Bevor Sie aber nun vor Begeisterung Ihren Lautstärkeregler in die rechte Anschlagposition bringen und den Staub von den Lautsprechermembranen schütteln, sollten Sie auf die Uhr sehen. Ein Kopfhörer ist vielleicht der günstigere Schallwandler. (hm)

Alter Joystick am C 16

Inkompatibilität bestimmt den C 16 – auch bei den Joystickports. Mit ein wenig Geschick läßt sich jedoch ein Adapter herstellen.

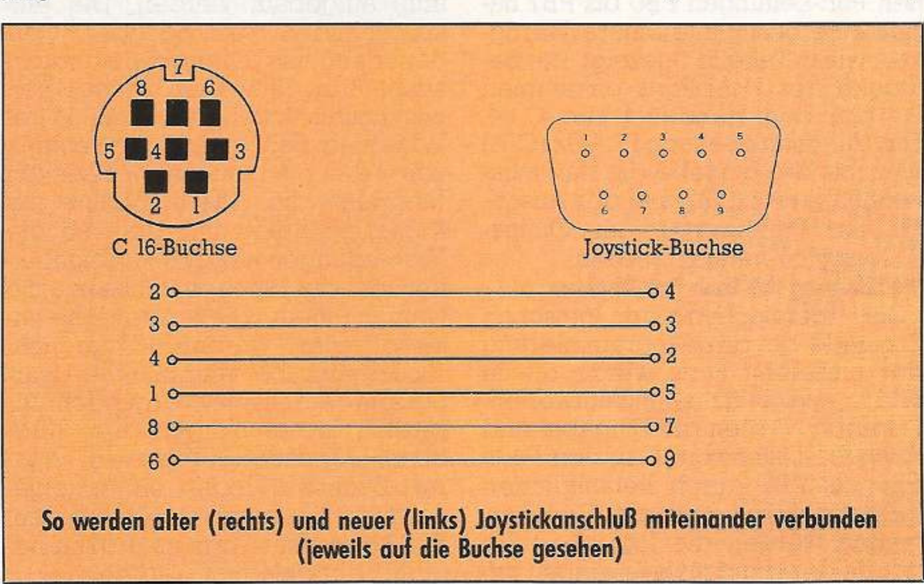
Ein recht geschickter Schachzug ist Commodore mit seinen Joystickports am C 16 gelungen. Denn bisher vertreibt nur Commodore die passenden Joysticks. Und diese bestechen nun wirklich nicht gerade durch ihre mechanische Qualität.

Wer nun einen guten, aber teureren Joystick, oder ganz einfach einen älteren am C 16 benutzen will, der benötigt einen Adapter. Eine preiswerte Lösung ist der Selbstbau. Die Herstellung ist recht einfach:

Sie brauchen nur etwas Kabel und zwei Stecker sowie einen 9-Pol-Canon-Stecker für den Joystick und einen 8-Pol-Micro-Din-Stecker für den Computer.

Beide Stecker erhalten Sie in jedem guten Elektronikfachgeschäft. Die beiden Stecker müssen Sie dann nur gemäß der Abbildung verbinden, und schon können Sie alle anderen handelsüblichen Joysticks am C 16 verwenden.

(Peter Schneider/ev)



So werden alter (rechts) und neuer (links) Joystickanschluß miteinander verbunden (jeweils auf die Buchse gesehen)

User-Port-Display

Eine kleine elektronische Schaltung setzt den User-Port ins richtige Licht. Sie zeigt auch, wie der User-Port für Steuerungsaufgaben eingesetzt werden kann.

Die Computer VC 20 und C 64 haben als Verbindung zur Außenwelt an der Gehäuse-Rückseite den User-Port. Dieser 8 Bit breite parallele Port besteht aus acht programmierbaren Anschlüssen PB0 bis PB7, die als Eingabe- oder Ausgabeleitung geschaltet werden können. Weiterhin stehen noch die Eingabe-Handshakeleitung FLAG2 (CA1 beim VC 20) und die Ausgabe-Handshakeleitung PA2 (VC 20: CB2) für diese acht Leitungen zur Verfügung.

Diese Anschlüsse sind vom User-Port aus mit dem universellen Interface-Baustein (VIA) 6522 direkt verbunden (VIA = Versatile Interface Adapter). Sie erlauben den Anschluß von externen Schaltungen und Geräten. Das User-Port-Display, das noch als universeller Ein-/Ausgabe-Adapter verwendet werden kann, wird an dieser Platinen-Steckerleiste am besten über einen Platinenstecker der Bezeichnung 251-12-50-170 angeklemmt. Die einzelnen Port-Leitungen PB0 bis PB7 haben eine bestimmte binäre Wertigkeit, wie in Tabelle 1 gezeigt. Die Belegung des User-Ports entnehmen Sie bitte dem Handbuch zum C 64. Die Handshakeleitung FLAG2 (CA1) kann bei der Eingabe, die Handshakeleitung PA2 (CB2) bei der Ausgabe von Datenwörtern als Quittierungssignal benutzt werden.

Die Schaltung des User-Port-Displays

Zur Überwachung der logischen Zustände bei der Programmierung der einzelnen Ports wurde die in Bild 1 gezeigte Schaltung entworfen. Dadurch werden die Eingabe- und Ausgabe-Datenwörter an den Ports PB0 bis PB7 durch Leuchtdioden (LEDs) sichtbar gemacht. Gleichermaßen werden die logischen Zustände der Handshakeleitungen angezeigt. Eine Stückliste für den I/O-Adapter (Input/Output) finden Sie in Tabelle 2. High-Pegel, also logisch »Eins«, werden durch Leuchten, Low-Pegel (logisch »Null«) durch Verlöschen der entsprechenden LED signalisiert.

Die Schaltung besteht aus zehn Invertern (2 x 74LS04, Bild 2), an die über je einen Widerstand von 330 Ohm die LED an +5 V angeschlossen sind. Ein Inverter kehrt das Eingangssignal um. Aus +5 V werden 0 V, aus 0 V werden +5 V.

Spannungsversorgung und Aufbau

Zur Spannungsversorgung für die Anzeigeeinheit wird die an Pin 2 des User-Port anliegende Spannung von +5 V benutzt. Es ist darauf zu achten, daß der Stromverbrauch der gesamten Anordnung 100 mA nicht übersteigt. In Fällen größerer Stromentnahme wird ein separates Netzteil erforderlich.

Port	Wertigkeit	binär
PB0	1	2^0
PB1	2	2^1
PB2	4	2^2
PB3	8	2^3
PB4	16	2^4
PB5	32	2^5
PB6	64	2^6
PB7	128	2^7

Tabelle 1. Die binären Wertigkeiten der Ein-/Ausgabeleitungen

Aufbau der Schaltung

Da es sich bei dieser Schaltung um eine relativ einfach aufzubauende Einheit handelt, wurde auf ein Platinenlayout verzichtet. Statt dessen kann die Schaltung auf eine Lochrasterplatte (Rastermaß: 2,51 mm) aufgebaut werden. Die entsprechenden Verbindungen werden dann mit Drahtverbindungen ausgeführt. Entweder durch Löten von Drahtstücken, oder mit Wire-Wrapping. Bei der letzteren Technik umwickelt man mit einer speziellen Maschine die Anschlußbeine mit Kupferlackdraht und stellt so die Verbindungen her. Die Anzeigeeinheit und die 13polige Buchsenleiste kann in einem kleinen Gehäuse untergebracht werden. Über ein 12adriges Kabel wird nun der User-Port mit dem Display verbunden. Zusätzlich ist es sinnvoll, die +5-V-Spannung und die Masse auf separate Buchse zu legen, um ein eigenes Netzteil anschließen zu können. Pin 2 des User-Ports muß dann natürlich abgeklemmt werden.

Die Programmierung des User-Ports

Wie bereits erwähnt, lassen sich die acht Port-Leitungen als Ein-/Ausgabeleitungen programmieren. Nach dem Einschalten des Computers werden alle Ports automatisch auf Eingabe geschaltet. Dies läßt sich leicht daran erkennen, daß alle acht LEDs des User-Port-Displays leuchten. Über das Datenrichtungsregister, Adresse DDRB = 56579 (VC 20 = 37138), lassen sich die einzelnen acht Ports so programmieren, daß sie entweder als Ein- oder Ausgabeleitung arbeiten. Mit dem Befehl POKE DDRB,X wird die Programmierung vorgenommen. Hierbei ist »X« eine Dezimalzahl zwischen 0 und 255. Die der Dezimalzahl entsprechende Binärzahl erscheint dann am User-Port an den Ausgängen PB0 bis PB7. In der Binärzahl stellt die logische »0« einen Eingabe-, die logische »1« einen Ausgabeport dar. Als logische Pegel liegen allerdings die invertierten Pegel an, da der C 64 und VC 20-Computer mit negativer Logik arbeitet; also High-Pegel bei logisch »0« und umgekehrt.

Mit POKE DDRB,255 werden alle acht Ports zu Ausgabeleitungen, da dezimal 255 in Binärdarstellung 1111111 ist. Die Programmierung mit POKE DDRB,0 ergibt entsprechend 00000000 und alle acht Ports sind Eingabeleitungen. POKEt man in die Speicherstelle DDRB die Dezimalzahl 240, so ergibt dies die Binärzahl 11110000. Dies bedeutet, daß die Ports PB0 bis PB3 Eingabeleitungen und die Ports PB4 bis PB7 Ausgabeleitungen sind. So lassen sich durch POKEn einer Zahl zwischen 0 und 255 in die Speicherstelle DDRB die jeweils gewünschten Ein-/Ausgabekombinationen einstellen.

Die Ausgabe von Daten

Nachdem das Datenrichtungsregister auf Ausgabe programmiert ist, können die Daten über die Speicherstelle PRB = 56577 (VC 20: PRB = 37136) ausgeben werden. Der entsprechende Befehl hierzu lautet POKE PRB,X. Auch hier kann X eine Dezimalzahl zwischen 0 und 255 sein. Die entsprechende Binärzahl erscheint an den Ports und bringt die dazugehörigen LEDs zum Leuchten. High-Pegel bedeutet jetzt Aufleuchten der LED. Mit Listing 1 kann dies leicht überprüft werden. In Zeile 10 werden alle Ports PB0 bis PB7 auf Ausgabe geschaltet. In Zeile 20 erfolgt die Zahleneingabe (Zahl zwischen 0 und 255). In Zeile 30 wird

diese Zahl in die Speicherstelle PRB gePOKEt und an den Ports als Binärzahl ausgegeben. Gleichzeitig stehen diese Signale an der 13poligen Buchsenleiste an, wo Interfaces für die vielfältigsten Schalt- und Kontrollaufgaben angeschlossen werden können.

Die Ausgabe-Handshakeleitung

Mit der Ausgabe-Handshakeleitung PA2 (VC 20: CB2) kann man die Ausgabe von Datenwörtern zu Steuerzwecken quittieren. So kann ein angeschlossenes Gerät durch diese Handshakeleitung zur Übernahme der Daten freigegeben werden.

Mit POKE 37148,220 wird an Port CB2 des VC 20 Low-Pegel, mit dem Befehl POKE 37148,255 High-Pegel gelegt. Beim C 64 sind dies die Befehle POKE 56576,147 für Low-Pegel und POKE 56576,151 für High-Pegel an Port PA2. Eine spezielle Anwendung bleibt dem Benutzer überlassen. Der Befehl kann in einem Programm an die Stelle eingefügt werden, wo er eine Schaltfunktion übernehmen soll. Listing 2 soll die Wirkungsweise von CB2 und PA2 veranschaulichen.

In Zeile 10 werden die Ports auf Ausgabe programmiert. In Zeile 20 wird die auszugebende Binärzahl als Dezimalzahl in die Speicherstelle PRB geschrieben. In Zeile 30 erfolgt dann der Quittungsbefehl. In diesem Fall wird ein Low-Pegel am Port PA2 oder CB2 ausgegeben. Nach dem Durchlauf einer Warteschleife in Zeile 40 wird PA2 (CB2) wieder auf High-Pegel zurückgesetzt, und die grüne LED leuchtet.

7404
7405
7406
7416

Bild 2. Die Pin-Belegung eines SN 7404

Eingabe von Daten

Das Datenrichtungsregister wird zu diesem Zweck mit POKE DDRB,0 auf Eingabe programmiert. Dies wird deutlich durch das Aufleuchten der LED angezeigt. Die entsprechenden Ports können nun auf Low-Pegel gelegt werden und bilden hierdurch eine bestimmmte Binärkombination. Mit den LEDs läßt sich die eingestellte Binärzahl anschaulich ablesen. Über den Befehl PRINT PEEK(PRB) wird der logische Zustand an den Ports abgefragt und auf dem Bildschirm erscheint die Zahl 189.

Das nachstehende Programm verdeutlicht noch einmal diese Möglichkeit der Dateneingabe:

```
10 POKE DDRB,0
20 PRINT "Ausgabe des Datenwortes"; PEEK(PRB)
30 GOTO 20
```

In Zeile 10 werden die Ports auf Eingabe programmiert. In Zeile 20 werden die angelegten Daten eingelesen und auf dem Bildschirm angezeigt.

Die Eingabe-Handshake-Leitung

Mit einem weiteren Befehl kann der Computer veranlaßt werden, das Basic-Programm anzuhalten und zu warten, bis eine Dateneingabe erfolgt ist. Hierzu ist die Eingabe-Handshake-Leitung CA1 FLAG2 (VC 20: CA1) als Quittungssignal vorhan-

1	Steckerleiste 251-12-50-170
1	13pol. Buchsenleiste
2 m	12adriges Kabel
2	IS 74LS04
8	LED rot (3 mm ∅)
1	LED grün (3 mm ∅)
1	LED gelb (3 mm ∅)
10	Widerstände 330 Ω/0,25 W
1	Lochrasterplatte
1	Miniaturbuchse für +5 V (extern)
1	Miniaturbuchse für GND (extern)
1	TEKO-Gehäuse (Typ 2A)

Tabelle 2. Die Stückliste zum User-Port-Display

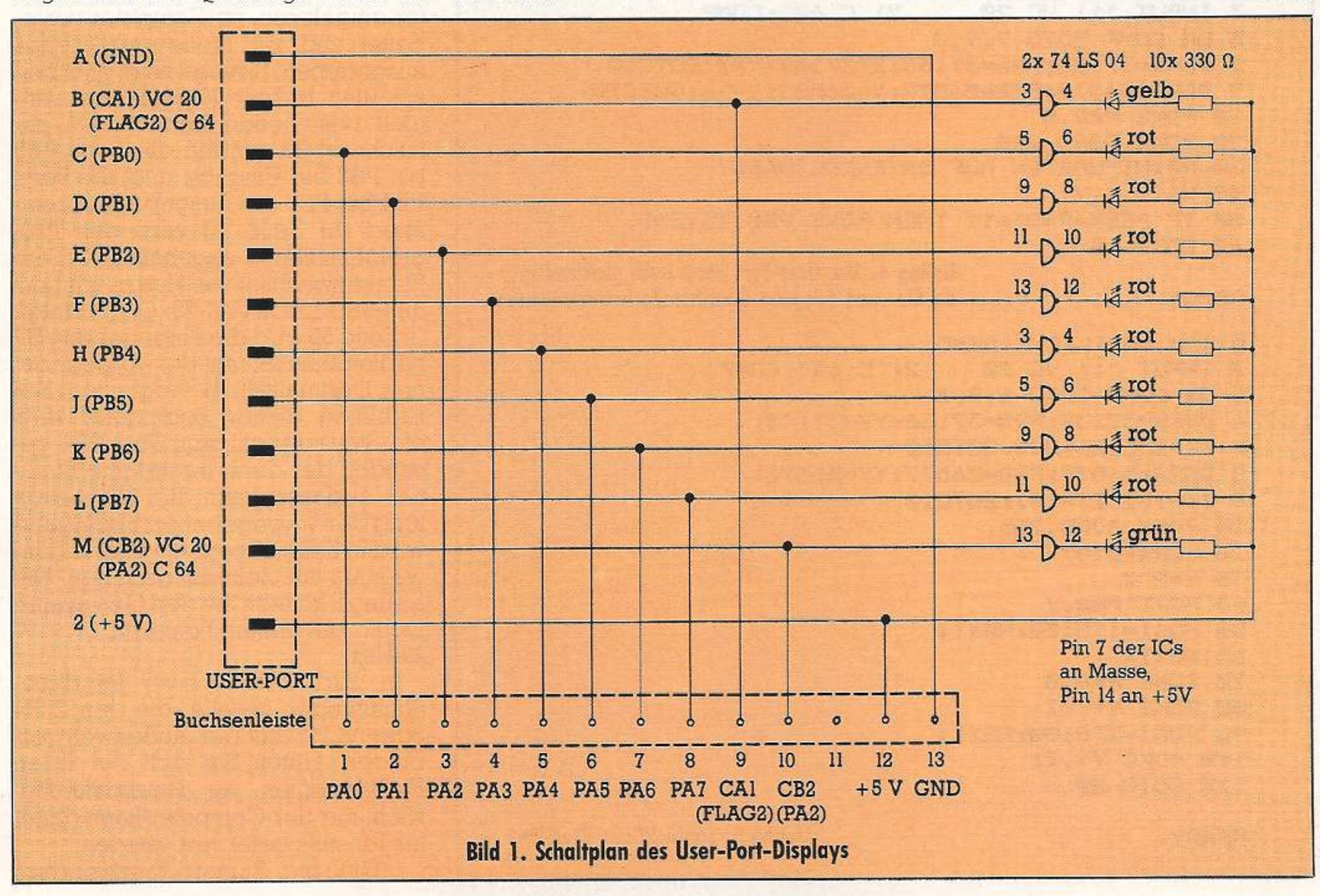


Bild 1. Schaltplan des User-Port-Displays

```
0 REM LISTING 1
3 REM VC 20:DDRB=37138 / C 64:DDRB=56579
5 REM VC 20: PRB=37136 / C 64: PRB=56577
10 POKE DDRB,255
20 INPUT "ZAHL";X
30 POKE PRB,X
40 GOTO 20

READY.
```

Listing 1. Ausgabe einer Binärzahl am User-Port

```
0 REM LISTING 2      C 64
5 DDRB=56579:PRB=56577
10 POKE DDRB,255
20 POKE PRB,127
30 POKE 56576,147
40 FOR I=1 TO 100:NEXT I
50 POKE 56576,151

READY.
```

Listing 2a. Demonstration der Handshakeleitung beim C 64

```
0 REM LISTING 2      VC 20
5 DDRB=37138:PRB=37136
10 POKE DDRB,255
20 POKE PRB,127
30 POKE 37148,220
40 FOR I=1 TO 100:NEXT I
50 POKE 37148,225

READY.
```

Listing 2b. Demonstration der Handshakeleitung beim VC 20

```
0 REM LISTING 3
3 INPUT "1) VC 20       2) C 64";COMP
5 ON COMP GOTO 7,9,3
7 DDRB=37138:PRB=37136:X=37149:Y=2:GOTO10
9 DDRB=56579:PRB=56577:X=56589:Y=16:GOTO10
10 POKE DDRB,0
20 PRINT"WARTEN AUF DATENFREIGABE"
30 WAIT X,Y
40 PRINT"AUSGABE DES DATENWORTES";PEEK(DRB)
50 GOTO20

READY.
```

Listing 3. So einfach ist das Einlesen mit Handshake von Daten am User-Port

```
0 REM LISTING 4
3 INPUT "1) VC 20       2) C 64";COMP
5 ON COMP GOTO 7,9,3
7 DDRB=37138:PRB=37136:X=37149:Y=2:GOTO10
9 DDRB=56579:PRB=56577:X=56589:Y=16:GOTO10
10 POKE PRB,0
20 POKE DDRB,240
30 PRINT"WARTEN AUF DATENFREIGABE"
40 WAIT X,Y
50 IF PEEK(PRB)=11 THEN POKE PRB,16:END
60 GOTO 50

READY.
```

Listing 4. Der User-Port kann auch gleichzeitig als Ein- und Ausgabe-Schnittstelle benutzt werden

```
0 REM LAUFLICHT-DEMO
3 INPUT "1) VC 20      2) C 64";COMP
5 ON COMP GOTO 6,8,3
6 DDRB=37138:PRB=37136:YY=37148:
7 ZZ=255:Z1=220:GOTO10
8 DDRB=56579:PRB=56577:YY=56576:
9 ZZ=151:Z1=147:GOTO10
10 POKE DDRB,255
20 FORX=0TO7
30 Y=2^X
40 POKE PRB,Y
50 FORI=1TO100:NEXTI
60 NEXTX
70 POKE PRB,0
80 POKE YY,ZZ
90 FORI=1TO100:NEXTI
100 POKE YY,Z1
110 GOTO 20

READY.
```

Listing 5. Lauflicht am User-Port

den. Bei einem Flankenwechsel von High nach Low werden die anliegenden Daten eingelesen. Mit Hilfe von Listing 3 läßt sich dieser Vorgang bewerkstelligen.

In Zeile 10 werden die Ports auf Eingabe programmiert. In Zeile 30 wird das Basic-Programm solange angehalten, bis der Flankenwechsel am Port FLAG2 (CA1) erfolgt. Danach wird das Programm wieder fortgesetzt, und in Zeile 40 das angelegte Datenwort auf dem Bildschirm angezeigt.

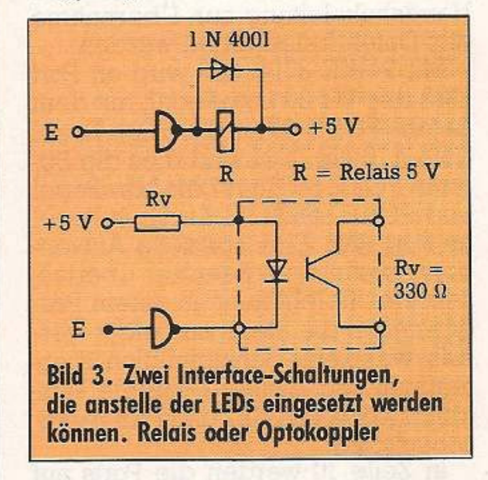


Bild 3. Zwei Interface-Schaltungen, die anstelle der LEDs eingesetzt werden können. Relais oder Optokoppler

Daten-Ein- und -Ausgabe kombiniert

Mit der Kombination von Daten-Ein- und Ausgabe können vielfältige Steuer- und Kontrollaufgaben gelöst werden. Hierbei werden Datenwörter eingelesen, im Computer verarbeitet und zur Steuerung wieder ausgegeben. Listing 4 zeigt dies beispielhaft. In Zeile 10 wird eine eventuell bestehende Kombination gelöscht. In 20 werden die Ports PB0 bis PB3 auf Eingabe und die Ports PB4 bis PB7 auf Ausgabe programmiert. In Zeile 40 wird das Programm solange angehalten, bis ein Flankenwechsel von High nach Low an Port FLAG2 (VC 20: CA1) erfolgt. In Zeile 50 wird die eingegebene Binärkombination mit der vorgegebenen Dezimalzahl 11 verglichen. Bei Erfüllung dieser Bedingung wird ein Steuerbefehl zum Port PB4 gesendet, der dann irgendeine Funktion auslösen kann. Bei Nichterfüllung der vorgegebenen Bedingung wird in Zeile 60 eine Rücksprunganweisung zur Zeile 50 gegeben. Mit Listing 5 können Sie den C 64 in eine Lichtorgel umfunktionieren.

Ausblick

In Bild 3 sind zwei Interface-Schaltungen gezeigt, die den C 64 oder VC 20 mit der Außenwelt verbinden. Hüten Sie sich bei Ihren Experimenten vor Netzspannung. Nicht nur der Computer kann damit ins Jenseits befördert werden.

(Dipl.-Ing. Richard Kurzhals/hm)


64ER ONLINE

Dateiverwaltungen: Was Sie beim Kauf beachten sollten



Der Kauf einer Dateiverwaltung ist risikoreich. Denn die Schwächen eines solchen Programms tauchen oft erst auf, wenn schon eine große Anzahl von Daten eingegeben wurde. Der Entwickler des »Datenmanager 64« gibt wertvolle Ratschläge.



Bevor Sie losziehen und sich verschiedene Programme anschauen, erstellen Sie zuerst bitte ein »Mengengerüst«: Überlegen Sie sich, wie lang ein Datensatz Ihrer Datei (zum Beispiel eine Adresse) maximal sein kann, aus wievielen Feldern er besteht (Name, Vorname, PLZ ...) und wieviele Datensätze Ihre Datei — planen Sie großzügig ein oder zwei Jahre voraus — aufnehmen muß. Seien Sie bei der Erstellung dieses Mengengerüstes nicht zu knauserig. Nichts ist schlimmer, als nach einem Jahr feststellen zu müssen, daß die Datei zu groß für die vorhandene Dateiverwaltung wird.

Sortieren Sie nun die Programme aus, die dieses Mengengerüst nicht erfüllen. Die Werbung ist dabei nicht sehr hilfreich, da viele Firmen einen kleinen Trick verwenden: Sie geben zum Beispiel an, daß Ihr Programm maximal 2000 Datensätze verwalten kann und daß ein Datensatz maximal 250 Zeichen lang sein darf. Nach dem Kauf stellt sich dann oft heraus, daß sich beide Versprechungen gegenseitig ausschließen: Entweder kann das Programm 2000 Datensätze verwalten, jedoch nur mit einer Länge von 70 Zeichen, oder aber es kann Datensätze mit einer Länge von 250 Zeichen verwalten, davon jedoch keine 2000.

Ein besserer Weg ist es, sich direkt beim Fachhandel zu erkundigen. Angesichts der Programmvielfalt für einen Computer wie den C 64, sind jedoch nur wenige Verkäufer in der Lage, detaillierte Auskunft über einzelne Programme zu geben. Als Lösung verbleibt somit nur ein intensives Studium von Zeitschriften und den darin erschienenen Tests.

Nachdem Sie nun diejenigen Programme aussortiert haben, die Ihre (zukünftigen) Datenmengen nicht verkraften können, kommt die zweite Selektionsstufe: die Frage der Datensicherheit. Dieses Kriterium wird oftmals unterschätzt. Viele Benutzer achten nur auf die Leistungsfähigkeit eines Programms. Ebenso wichtig ist jedoch, mit welcher Sicherheit und Anwenderfreundlichkeit diese Leistung erbracht wird.

»Verdächtige« Programme meiden

Hören Sie sich bei Ihren Bekannten um, und lesen Sie die Tests in den entsprechenden Zeitschriften. Sollte es Hinweise für die mangelnde Datensicherheit eines Programms geben, so streichen Sie dieses unverzüglich aus Ihrer Liste.

Sollten Sie einen Drucker besitzen, ist es natürlich selbstverständlich, daß das jeweilige Programm diesen unterstützen muß. Wenn Sie, zusammen mit einem Interface, den Drucker eines Fremdherstellers verwenden, so testen Sie die Kompatibilität mit den in Frage kommenden Programmen. Sollten Sie keinen Drucker besitzen, erkundigen Sie sich, ob über den Bildschirm allein alle Programmfunktionen ausgenutzt werden können.

Weitere notwendige Eigenschaften:

◆ Es muß möglich sein, mindestens ein frei wählbares »Schlüssel-« oder »Indexfeld« anzugeben. Ein solcher Schlüssel erlaubt die schnelle Suche nach Datensätzen auch bei umfangreichen Dateien.

◆ Alle Felder des Datensatzes müssen bei der Suche durch »UND« miteinander verknüpft werden können. Beispiel: Selektiere alle Adressen mit dem Namen »Müller« **und** dem Wohnort »Mannheim«.

◆ Selbstverständlich muß die Datei nach einem frei wählbaren Feld des Datensatzes sortiert werden können.

Universell ist besser als speziell

Dateiverwaltungen, die die bisher geschilderten Anforderungen nicht erfüllen, sollten Sie von Ihrer Liste streichen. Auf einen weiteren Punkt sollten Sie jedoch noch achten: Speziell für den C 64 gibt es eine Fülle verschiedener Datenverwaltungen, mit denen Sie beliebige Arten von Dateien aufbauen können (Adreßdatei, Schallplattendatei etc.), aber auch Programme, die nur eine bestimmte Dateiart verwalten, meist eine Adreßdatei — das wohl häufigste Anwendungsgebiet. Selbst wenn Sie momentan nur eine solche Adreßdatei verwalten wollen, so überlegen Sie bitte, ob es nicht dennoch sinnvoller ist, ein Programm zu kaufen, mit dem sich beliebige Arten von Dateien verwalten lassen. Betrachten Sie eine eventuelle Mehrausgabe als Investition in die Zukunft. Sie wissen heute vielleicht noch nicht, welche zukünftigen Aufgaben Sie für Ihren Computer noch finden werden.

Die zusätzlichen Kriterien, anhand derer eine Dateiverwaltung beurteilt werden kann, sind zu einem großen Teil abhängig von Ihren persönlichen Ansprüchen an den Komfort des Programms:

◆ Erlaubt das Programm die Verwendung mehrerer Indizes? Durch Verwendung eines Schlüsselfeldes

bei der Suche ist selbst eine schnelle Selektion bestimmter Datensätze möglich. Findet in dem jeweiligen Schlüsselfeld hingegen kein Eintrag statt — Sie wissen nur noch, daß die gesuchte Person in München wohnt, erinnern sich jedoch nicht an ihren Namen — so können Sie bei sehr umfangreichen Dateien nicht selten eine Zigarettenpause einlegen, bis der gewünschte Datensatz gefunden wird. Daher ist es vorteilhaft, wenn für mehrere Felder eines Datensatzes ein solcher Index angelegt werden kann.

◆ Sind außer der »UND«-Verknüpfung von Suchkriterien noch weitere Verknüpfungsmöglichkeiten vorhanden, zum Beispiel »ODER« (Selektiere alle Personen, die Müller heißen **oder** in Mannheim wohnen)?

◆ Möglicher weiterer Suchkomfort (und auch bei vielen Programmen verwirklicht): die abgekürzte Eingabe von Suchkriterien, das Suchen mit Hilfe von Vergleichsoperatoren (>, <, > =, < =), alle Adressen mit Postleitzahl zwischen 4450 und 4459)) oder »Ma?er« findet »Maier« und »Mayer«.

◆ Ist es möglich, außer einem primären Sortierkriterium noch ein sekundäres anzugeben oder eventuell gar noch weiter in die Tiefe zu sortieren (Beispiel: Sortiere nach »Name«. Sind zwei Namen gleich, sortiere diese nach »Vorname«)?

◆ Ausdruck: Prinzipiell sollten sich alle Möglichkeiten einer Dateiverwaltung auch ohne Drucker erschließen lassen. Achten Sie daher darauf, ob zum Beispiel die sortierte Ausgabe von Datensätzen auch auf dem Bildschirm möglich ist. Sollten Sie jedoch einen Drucker besitzen, ist für Sie wichtig, wie frei Sie den Ausdruck gestalten können. Mindestanforderungen: Sie müssen angeben können, ob alle oder nur spezifische Felder eines Datensatzes gedruckt werden sollen, ob die Felder eines Datensatzes neben- oder untereinander stehen sollen, wie groß der Randabstand sein soll etc.. Einige Programme bieten dem Anwender sogar die Möglichkeit, eine eigene, völlig frei gestaltbare Druckmaske zu entwerfen. Wichtig ist auch, einen Datensatz jederzeit vom Bildschirm ohne »Verrenkungen« auszudrucken.

◆ Drei weitere Punkte dürften speziell für kommerzielle Anwender wichtig sein:

1. Textverarbeitung: Ist es möglich, mit den Adressen einer Adreßdatei Serienbriefe zu erstellen, oder muß ein weiteres Programm gekauft werden, und gibt es dieses Programm passend zur Dateiverwaltung?

2. Globale Operationen, zum Beispiel die Änderung der Preise aller Artikel in einer Artikeldatei um 10 Prozent in einem Arbeitsgang.

3. Verknüpfung von Dateien, zum Beispiel die Verknüpfung von Adreß- und Lagerdatei zum Schreiben von Rechnungen. Von echten Datenverwaltungen (im Gegensatz zu Datenbanken) können derartige dateiübergreifende Verknüpfungen jedoch nicht erwartet werden.

Erfüllung des Mengengerüstes und Datensicherheit sind unabdingbare Mindestanforderungen, wohingegen Suchkomfort, Textverarbeitung etc. weitgehend vom Geschmack (und Geldbeutel) des Einzelnen abhängig sind. Ein äußerst wichtiges Kriterium zur Beurteilung einer Dateiverwaltung habe ich mir jedoch bis zum Schluß aufgespart: die Zugriffsgeschwindigkeit. Wenn ein Programm Dateien mit 50 Datensätzen schnell verwaltet, sagt dies nichts über die Leistungsfähigkeit des gleichen Programms mit 500 Datensätzen aus. Das Verhalten bei großen Datenmengen ist ein sehr komplexes Problem, das von den verwendeten Datenstrukturen abhängt und sehr unterschiedlich sein kann. Ideal wäre es, wenn jeder Hersteller eines Programms zu diesem »vollgepackte« Demo-Dateien anbieten würde, die ein potentieller Käufer beim Fachhändler testen kann. Da dies im Moment jedoch nur Wunschdenken ist, empfehle ich Ihnen das Studium von Tests oder aber das Umhören im Bekanntenkreis.

Als letzter Schritt erwartet Sie nun der Gang zum Fachhändler. Bei der Beurteilung einer Dateiverwaltung wird zumeist nur der Aspekt der Leistungsfähigkeit gesehen. Oft besteht jedoch ein »umgekehrt proportionaler« Zusammenhang zwischen der Leistungsfähigkeit und der Bedienungsfreundlichkeit eines Programms. Dies ist verständlich, denn je größer die Vielfalt an Funktionen ist, desto schwieriger wird es, diese dem Benutzer einfach zu »servieren«. In vielen Fällen muß sich der Anwender durch eine Unmenge von Menüs, Untermenüs und Unter-Untermenüs durchkämpfen, um eine bestimmte Funktion anzuwählen. Nehmen Sie sich daher außer Zeit noch Ihre Liste mit den in Frage kommenden Programmen, gehen Sie zum Händler, und probieren Sie möglichst viele Funktionen selbst durch. Begnügen Sie sich nicht mit eventuell vorbereiteten Demo-Versionen, sondern bauen Sie selbst Dateien auf. Nur auf diese Weise können Sie feststellen, ob ein Programm Ihnen nicht nur von seiner Leistung her zusagt, sondern auch vom Bedienungskomfort. Ein Programm, zu dessen Benutzung das wochenlange Studium von Handbüchern notwendig ist, sollten Sie sich nur dann kaufen, wenn Sie dessen Funktionen auch wirklich benötigen. (Said Baloui/gk)

Checkliste

Minimalforderungen:

- ☐ Wieviele Datensätze der von Ihnen benötigten Länge kann das Programm verwalten (beachten Sie ein mögliches Wachstum Ihrer Dateien in späteren Jahren)?
- ☐ Wieviele Felder darf ein Datensatz enthalten?
- ☐ Werden verschiedene Datentypen unterstützt?
- ☐ Was können Sie über die Datensicherheit in Erfahrung bringen?
- ☐ Kann für ein beliebiges Feld ein Index zur schnellen Suche erstellt werden?
- ☐ Ist ein minimaler Suchkomfort vorhanden (Verknüpfung von Suchkriterien, Abkürzen, Mengenoperatoren)?
- ☐ Können alle Funktionen ohne Drucker genutzt werden?
- ☐ Wird der eigene Drucker unterstützt?
- ☐ Ist das Sortieren der Datei möglich?
- ☐ Wie verhält sich das Programm bei großen Datenmengen (Zugriffsgeschwindigkeit und Datensicherheit)?
- ☐ Wie bedienungsfreundlich ist das Programm?

Zusätzlicher Komfort:

- ☐ Welche zusätzlichen Suchmöglichkeiten gibt es (Maskieren, Intervallsuche)?
- ☐ Können mehrere Schlüssel definiert werden?
- ☐ Wie frei ist die Gestaltung des Ausdrucks?
- ☐ Können mehrere Sortierkriterien verwendet werden?
- ☐ Sind globale Änderungs- oder Löschfunktionen vorhanden?
- ☐ Ist die Serienbrieferstellung möglich oder zumindest der Zukauf eines geeigneten Textprogramms?
- ☐ Können einzelne Dateien miteinander verknüpft werden?

Die wichtigsten Begriffe der Dateiverwaltung

Datensatz, Datensatzfeld:
Ein Datensatz ist eine Menge zusammengehöriger Informationen (zum Beispiel Daten eines Artikels: Artikelnummer, Preis etc.). Die einzelnen Elemente des Datensatzes werden Felder genannt (Bild 1).

Datei:
Eine Menge gleichartiger Datensätze (zum Beispiel alle Artikel, die Sie führen) wird als Datei bezeichnet (Bild 1).

Dateiverwaltung:
Programm, das eine Datei verwalten kann, das heißt, das in der Lage ist, vom Benutzer vorgenommene Anfragen zu beantworten (»Welche Adresse hat Herr Gustav Werner?«) und den Änderungsdienst zu erledigen (Eintragen neuer Datensätze, Ändern von Datensätzen, Löschen von Datensätzen).

Benutzerschnittstelle:
Unter Benutzerschnittstelle versteht man die Art und Weise, in der die Kommunikation zwischen Dateiverwaltung und Benutzer erfolgt, in der dieser zum Beispiel Anfragen an das Programm stellt, oder den sortierten Ausdruck der Datei veranlaßt. Die zwei gebräuchlichsten Arten der Benutzerschnittstelle sind die Menüsteuerung und die Steuerung mit Hilfe von Abfragesprachen.

Menüsteuerung:
Die Benutzung geschieht interaktiv, im Dialog mit dem Programm.

Abfragesprachen:
Der Benutzer formuliert seine Wünsche an die Dateiverwaltung mit Hilfe einer eigens dafür zur Verfügung gestellten Sprache; er schreibt gewissermaßen ein Abfrageprogramm.

Datenstrukturen:
Man unterscheidet logische und physische Datenstrukturen. Logische Datenstrukturen definieren die Beziehung der einzelnen Elemente einer Datei zueinander, physische Datenstrukturen entscheiden über die Art und Weise, in der die Daten auf einem Speichermedium abgelegt werden.

Sequentieller Zugriff:
Zugriffsart, bei der auf die Daten einer Datei nur nacheinander in der Reihenfolge der Abspeicherung zugegriffen werden kann. Um den x-ten Datensatz zu lesen, müssen erst die Datensätze 1, 2, ..., x-1 gelesen werden. Häufig verwechselt werden »sequentieller Zugriff« und »sequentielle Datei«: eine sequentielle Datei ist eine Datei, deren einzelne Elemente sich lückenlos hintereinander auf dem Speichermedium befinden (im Gegensatz beispielsweise zur Hashing-Datei). Die Art der Datei erlaubt jedoch noch keine Schlußfolgerung auf die Zugriffsart.

Direktzugriff:
Durch Angabe des Ortes, an dem sich ein Datensatz befindet (Floppy: Spur und Sektor) kann auf diesen direkt zugegriffen werden (Bild 2). Eine Sonderform des Direktzugriffs ist der relative Zugriff: Hierbei wird das Speichermedium in »Records« unterteilt, deren Länge beim Aufbau der Datei festgelegt wird und die der maximalen Datensatzlänge entspricht. Jeder Datensatz wird in einem solchen Record abgespeichert. Unter Angabe der jeweiligen Recordnummer kann auf jeden Datensatz der Datei direkt zugegriffen werden.

Schlüssel/Index:
Eine Datei wird nach einem oder mehreren Schlüsseln geordnet. Ein solcher Schlüssel ist meist ein Feld des Datensatzes, zum Beispiel »Name« oder »Artikelnummer«.

Schlüssel-/Indexdatei:
Eine Datei, die zu jedem Datensatz das Schlüsselkriterium enthält, nach dem die Datensätze geordnet sind, sowie einen Zeiger auf den eigentlichen Datensatz in der Datensatzdatei.

Index-sequentielle Datei:
Dateiform, bei der außer der Datensatzdatei noch eine Schlüsseldatei existiert. Da diese Schlüsseldatei von jedem Datensatz nur einen Teil enthält (zum Beispiel den Namen), ist sie klein im Vergleich zur Datensatzdatei und kann dadurch in den Computerspeicher geladen und in

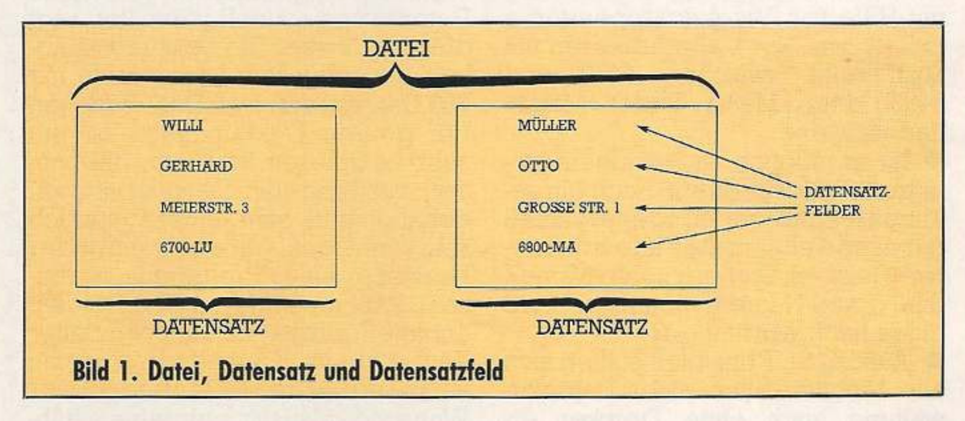


Bild 1. Datei, Datensatz und Datensatzfeld

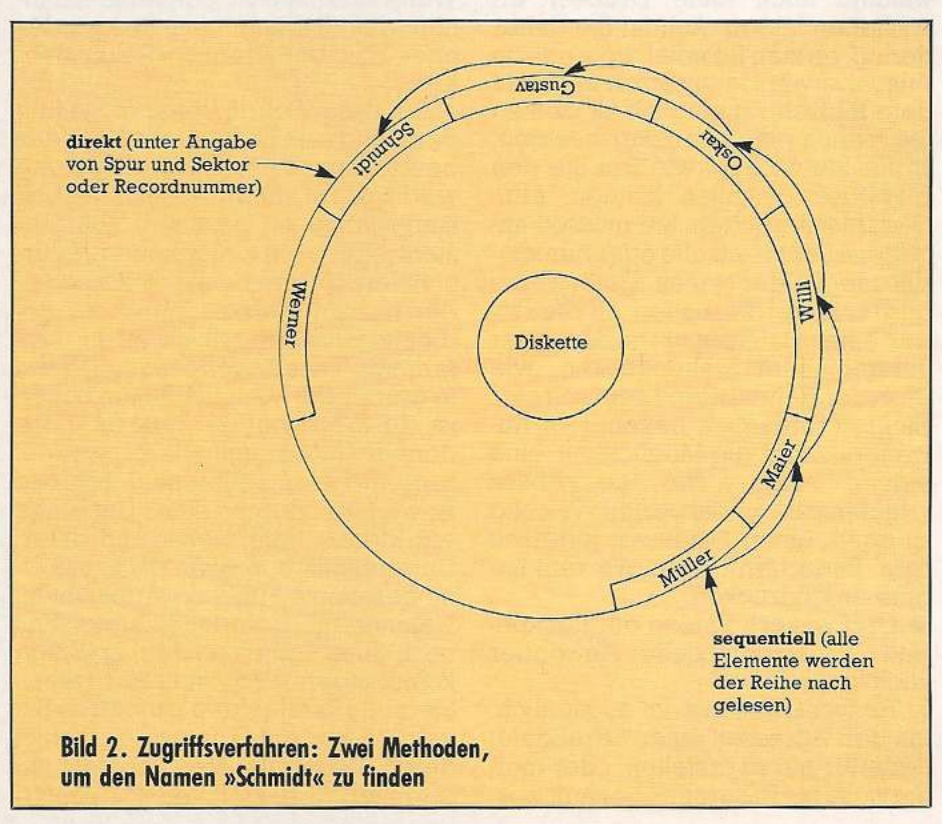


Bild 2. Zugriffsverfahren: Zwei Methoden, um den Namen »Schmidt« zu finden

diesem erheblich schneller als auf einem externen Speichermedium durchsucht werden (Bild 3). Wenn die Suche nach einem Datensatz nicht nach dem Schlüssel erfolgt, geht der Vorteil der schnellen Suche über die Indexdatei verloren; die Datensatzdatei muß sequentiell durchsucht werden (außer wenn das Dateiverwaltungsprogramm die Verwendung mehrerer Schlüssel gestattet).

Ordnung:
Die Ordnung einer Datei nach einem bestimmten Kriterium kann durch sequentielle Reihung entsprechend der Ordnung (einfache Liste), oder durch Verknüpfung der Daten über Zeiger (verkettete Listen, Baumstrukturen) geschehen.

Zeiger (Pointer):
Um Ordnung in eine Datei zu bringen, deren Elemente nicht in der Reihenfolge dieser Ordnung abgespeichert sind, verwendet man für jedes Element der Datei einen oder mehrere Zeiger, die auf den Speicherort des in der Reihenfolge nächsten oder auch vorhergehenden Elements zeigen.

Binäre Suche:
Eine — zum Beispiel alphabetisch oder numerisch — geordnete Datei läßt sich mit Hilfe der binären Suche sehr schnell durchsuchen: Zuerst wird auf das mittlere Element der Datei zugegriffen. Ist das gesuchte Element größer, anschließend auf die Mitte der rechten Hälfte der Datei und so weiter. Bei jedem Suchschritt wird die Länge der noch zu durchsuchenden Datei halbiert.

Hashing:
Ein Verfahren, bei dem der Ort, an dem ein Datensatz abgespeichert wird, mit Hilfe eines geeigneten Algorithmus direkt aus dem Inhalt dieses Datensatzes ermittelt wird (zum Beispiel, indem die Quersumme der ASCII-Zeichen des ersten Datensatzfeldes als Recordnummer beim relativen Zugriff verwendet wird). Hashing-Verfahren zeichnen sich durch außergewöhnlich schnellen Datenzugriff aus. Nachteilig ist, daß die Daten so abgespeichert werden, wie sie eingegeben wurden, und daher nicht geordnet sind.

Baumstruktur:
Eine häufig verwendete Datenstruktur, bei der jedem Datensatz mehrere Zeiger zugeordnet werden, durch die die Ordnung der Datei hergestellt wird; zum Beispiel einen Zeiger auf das »nächstgrößere« Element und einen Zeiger auf das »nächstkleinere« Element (Binärbaum, Bild 4).

Sortierverfahren:
Eine der häufigsten Aufgaben einer Dateiverwaltung ist die sortierte Ausgabe von Daten. Bei einer Hashing-Datei muß die Datei in einem solchen Fall unbedingt sortiert werden, da die Datensätze nach keiner erkennbaren Ordnung abgelegt sind. Aber auch bei einer geordneten Datei ist die Sortierung immer dann unumgänglich, wenn eine sortierte Ausgabe nach einem anderen Ordnungskriterium gewünscht wird als jenem, das zum Aufbau zum Beispiel der Baumstruktur verwendet wurde. Es gibt eine Vielzahl von Sortierverfahren (Sortieren durch Auswahl, durch Vergleich, durch Mischen, siehe 64er, Ausgabe 4/85 ff, Effektives Programmieren).

Interne Sortierverfahren:
Interne Sortierverfahren (zum Beispiel Quicksort) können immer dann angewendet werden, wenn sich die zu sortierenden Elemente im RAM befinden.

Externe Sortierverfahren:
Externe Sortierverfahren sind notwendig, wenn der Arbeitsspeicher (RAM) zu klein ist, um alle zu sortierenden Elemente aufzunehmen. In diesem Fall müssen die Elemente direkt auf dem externen Speicher (Floppy/Platte) sortiert werden. Meist werden mit Hilfe interner Sortierverfahren auf dem externen Speicher mehrere kleine, vorsortierte Dateien erzeugt, die dann mit entsprechenden Sortierverfahren weiterverarbeitet werden.

Reorganisation (update):
Dateien müssen teilweise aktualisiert werden. Dieser Vorgang wird als Reorganisation bezeichnet. Eine solche Reorganisation ist zum Beispiel immer dann nötig, wenn der Benutzer die Struktur der Daten verändern will (zum Beispiel neue Felder hinzufügt oder löscht) oder es zu Programmabstürzen kam.

(Said Baloui/gk)

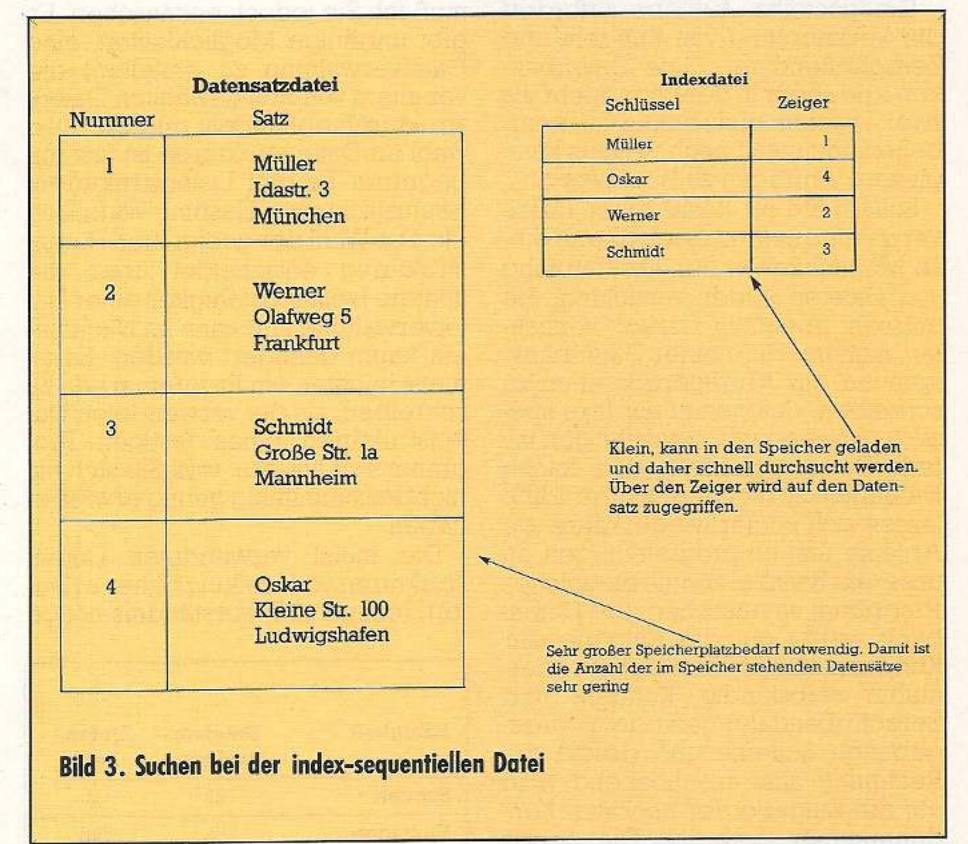


Bild 3. Suchen bei der index-sequentiellen Datei

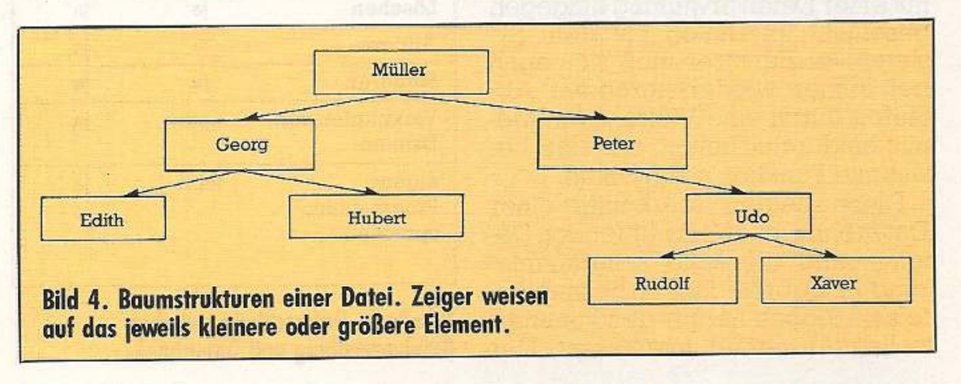


Bild 4. Baumstrukturen einer Datei. Zeiger weisen auf das jeweils kleinere oder größere Element.

Dateiverwaltung ist nicht gleich Datenbank

Alle reden über Dateiverwaltung. Über die vielfältigen Anwendungsmöglichkeiten und über Vor- und Nachteile der verschiedenen Programme. Wenig verbreitet ist jedoch das Wissen darum, was eine Dateiverwaltung eigentlich ist und welche Unterschiede zu Datenbanken bestehen.

Zuerst sei folgendes erklärt: eine Datenbank ist nicht einfach eine Art »bessere Dateiverwaltung«. Zwischen Datenbank und Dateiverwaltung bestehen nicht nur graduelle Unterschiede. Beiden gemeinsam ist die Aufgabe, eine Menge gleichartiger Daten, das heißt Dateien zu verwalten, ob es sich bei diesen Daten nun um Artikel, Schallplatten oder Videotitel handelt. Das Verwalten besteht im sogenannten »Änderungsdienst«, dem Ändern und Löschen von Datensätzen, dem Sortieren von Dateien und der möglichst komfortablen Beantwortung von Anfragen des Benutzers, also dem Selektieren bestimmter Datensätze aus der Gesamtmenge.

Mit einer leistungsfähigen Dateiverwaltung läßt sich nicht nur eine spezifische Datei verwalten, sondern der Benutzer kann beliebig viele verschiedene Dateien aufbauen. Mit einer Datenbank ist es jedoch möglich, mehrere Dateien miteinander zu verknüpfen (siehe Übersicht Tabelle 1).

Ein Beispiel: Stellen Sie sich vor, Sie seien der Inhaber eines Zeitungsvertriebs und seit kurzem stolzer Besitzer eines Computers. Diesen wollen Sie dazu verwenden, das Schreiben von Rechnungen an die Abonnenten zu automatisieren. Sie erwerben eine Datenbank wie zum Beispiel dBase II und bauen zwei Dateien auf: Eine Kundendatei, in der Sie Ihre Abonnenten führen (Kundennummer, Anschrift des Kunden, abonnierte Zeitschrift) und eine Zeitschriftendatei (Zeitschrift, Abonnentenpreis). Bisher verlief das Schreiben von Rechnungen so: Sie suchen in dem Karteikasten »Kunden« den jeweiligen Abonnenten, tragen dessen vollständige Anschrift in die Rechnung ein, merken sich den Namen der abonnierten Zeitschrift, suchen in dem Karteikasten »Zeitschriften« nach der entsprechenden Karteikarte und tragen den Abonnentenpreis in die Rechnung ein. Ein sehr zeitaufwendiger Vorgang, wenn die Dateien groß sind.

Der Datenbank müssen Sie nur angeben: »Suche in der Kundendatei nach dem Kunden xyz, merke dir die abonnierte Zeitschrift und suche in der Datei »Zeitschriften« nach diesem Titel. Die Datenbank verfügt nun über alle Informationen, um eine vollständige Rechnung auszudrucken.

Die gestellte Aufgabe erfordert die Verknüpfung von Kunden- und Zeitschriftendatei. Gute Datenbanken sind sogar in der Lage, mehr als zwei Dateien gleichzeitig zu bearbeiten und somit noch weitaus komplexere Anfragen zu beantworten.

Sollten Sie im Besitz einer Dateiverwaltung sein, müssen Sie auf diese Möglichkeiten der Verknüpfung von Dateien leider verzichten. Sie müssen ebenfalls darauf verzichten, sich mit Hilfe einer Datenbanksprache ein Abfrageprogramm zu schreiben, das genau auf Ihre spezielle Anwendung zugeschnitten ist. Jede Datenbank bietet eine solche Datenbanksprache. Mit Ihrer Hilfe lassen sich immer wiederkehrende Abläufe fest einprogrammieren. In unserem Beispiel könnte ein solches Programm so aussehen: die Datenbank wartet auf die Eingabe der Kundennummer, durchsucht daraufhin selbständig Kunden- und Zeitschriftendatei nach dem vorgegebenen Schema und druckt die Rechnung aus; anschließend wird auf die Eingabe der nächsten Kundennummer gewartet. Die Arbeit mit einer Dateiverwaltung hingegen geschieht im Dialog mit dem System. Der Benutzer muß sich auch bei immer wiederkehrenden Abläufen durch eine Vielzahl von Menüs hindurcharbeiten, um eine bestimmte Funktion anzuwählen.

Diese beiden Merkmale einer Datenbank, das Verknüpfen von Dateien, und die feste Programmierung immer gleicher Arbeitsabläufe sind jedoch nur für den kommerziellen Anwender interessant. Der private Benutzer, der seine Adressen oder Schallplatten verwaltet, wird mit einer leistungsfähigen Dateiverwaltung wohl immer zufrieden sein.

Sollten Sie aufgrund der Tatsache, daß Dateiverwaltungen nicht so leistungsfähig wie Datenbanken sind, nun der Ansicht sein, daß die Programmierung einer Dateiverwaltung wohl recht einfach sein müßte, muß ich Sie jedoch enttäuschen. Es gibt unzählige Möglichkeiten, eine Dateiverwaltung zu erstellen, die vor allem von den gewählten Datenstrukturen abhängen; und die Auswahl an Datenstrukturen ist fast unbegrenzt. Es gibt Listenstrukturen, Baumstrukturen, Hashing-Verfahren etc. Die Wahl der geeigneten Datenstrukturen entscheidet über die spätere Leistungsfähigkeit einer Dateiverwaltung und kann im Nachhinein kaum geändert werden. Es ist meist leichter, ein Programm neu zu schreiben, als die verwendeten Datenstrukturen eines fertigen Programms zu ändern, weil Sie sich als nicht leistungsfähig genug erwiesen haben.

Die meist verwendeten Datenstrukturen will ich kurz näher erläutern (einige zum Verständnis nötige

Fähigkeit	Dateiverwaltung	Datenbank
Suchen	ja	ja
Eintragen	ja	ja
Löschen	ja	ja
Ändern	ja	ja
Sortieren	ja	ja
Verknüpfen von Dateien	nein	ja
eigene Programmiersprache	nein	ja

Tabelle 1. Unterschiede zwischen Dateiverwaltung und Datenbank

Begriffe wie »binär«, »Zeiger« etc. sind im Lexikon in dieser Ausgabe erläutert). Sollte dieser Artikel bei Ihnen Appetit auf »mehr« wecken, so möchte ich Sie auf das Buch »Alles über Datenbanken und Dateiverwaltung mit dem C 64«, erschienen bei Data Becker, hinweisen. Kommen wir nun zu den versprochenen Datenstrukturen (Tabelle 2).

Listenstrukturen können Sie sich als eine Aneinanderreihung von Datensätzen vorstellen, vergleichbar mit der Anordnung von Daten in einem Telefonbuch. Diese Aneinanderreihung kann ungeordnet oder geordnet sein (meist alphabetisch). Geordnete Listen haben den Vorteil, daß sie schneller durchsucht werden können als ungeordnete. Sie können die Suche in einer geordneten Liste mit der Suche in einem Telefonbuch vergleichen, bei der Sie wohl kaum auf der ersten Seite mit der Suche nach »Maier« beginnen, sondern wesentlich effizienter suchen, eben (fast) binär. Problematisch an geordneten Listen ist das Eintragen und Löschen von Datensätzen. Um einen neuen Datensatz einzutragen, müssen alle alphabetisch nachfolgenden Datensätze verschoben werden, um Platz zu schaffen. Aus diesem Grund wird die Ordnung bei verketteten Listen nicht durch die Reihenfolge, sondern durch Zeiger hergestellt. Jedes Element der Liste besitzt einen solchen Zeiger, der auf den Ort weist, an dem der (alphabetisch) nachfolgende Datensatz abgespeichert ist. Da die Reihenfolge der Elemente unerheblich ist, kann ein neu einzutragender Datensatz einfach an das Ende der Liste angehängt werden. Er muß allerdings einen Zeiger auf den alphabetisch nachfolgenden Datensatz bekommen, und der vorhergehende Datensatz einen Zeiger auf den gerade neu eingetragenen. Eine verkettete Liste läßt sich leider nicht binär durchsuchen, sie muß sequentiell, also Element für Element, durchsucht werden. Dafür erlaubt sie das schnelle Eintragen und Löschen von Datensätzen. Da sie geordnet ist, kann sich der Benutzer einer solchen Dateiverwaltung jederzeit den alphabetisch nächsten oder vorhergehenden Datensatz zeigen lassen.

Baumstrukturen sind im Grunde genommen nur eine besondere Form von geordneten Listen, bei der die einzelnen Elemente, allerdings durch mehr als einen Zeiger, miteinander verkettet werden. Zum Einfügen eines neuen oder Löschen eines alten Elementes müssen nur wenige Zeiger verändert werden; das Verschieben von Elementen ist nicht nötig. Die Suche nach bestimmten Elementen ist schnell, da der Baum binär durchsucht wird. Aufgrund Ihrer Vorteile gehören Baumstrukturen zu den beliebtesten Datenstrukturen.

Hashing: Das sogenannte »Hash-Verfahren« hat mit den bisher geschilderten Datenstrukturen keinerlei Gemeinsamkeiten. Es beruht darauf, daß mit Hilfe eines geeigneten Algorithmus direkt aus dem abzuspeichernden Datensatz die Adresse gewonnen wird, an dem dieser abgelegt wird. Ein Beispiel: der einfachste denkbare Hash-Algorithmus bestünde darin, die Quersumme der einzelnen ASCII-Werte für ein einzelnes Datensatzfeld, zum Beispiel »maier«, zu bilden. Die ermittelte Zahl (366) könnte nun als Recordnummer oder als Index für einen String verwendet werden (a$(366)="maier;georg;münchen"). Bei der Suche nach einer bestimmten Adresse wird nun dieser Algorithmus auf den Namen angewandt, und der String mit diesem Index oder aber der jeweilige Record eingelesen. Der Vorteil des Hash-Verfahrens liegt in der Geschwindigkeit: in den meisten Fällen ist keinerlei Suche nach bestimmten Daten mehr nötig, sondern es kann direkt auf den gewünschten Datensatz zugegriffen werden. Nachteilig ist, daß die Daten völlig ungeordnet sind.

Diese kurze Vorstellung verschiedener Datenstrukturen sollte vor allem eines zeigen: Die Wahl einer bestimmten Datenstruktur hängt von der jeweiligen Aufgabenstellung ab. Wenn der entscheidende Gesichtspunkt bei der Verwaltung von Daten die Zugriffsgeschwindigkeit ist, empfiehlt sich das Hash-Verfahren. Muß die Datei jedoch unbedingt alphabetisch durchblättert werden können, so bieten sich Listen- oder Baumstrukturen an.

Struktur	Vorteile	Nachteile
ungeordnete Liste	einfacher Änderungsdienst (Blockoperationen nur zum Löschen nötig)	langsame sequentielle Suche, kein geordnetes Blättern
geordnete Liste	schnelle binäre Suche, geordnetes Blättern	aufwendiger Änderungsdienst (Blockoperationen)
verkettete Liste	geordnetes Blättern, einfacher Änderungsdienst (keinerlei Blockoperationen)	langsame sequentielle Suche
Baumstrukturen	schnelle binäre Suche, geordnetes Blättern, einfacher Änderungsdienst (keinerlei Blockoperationen)	—
Hashing	extrem schnelle Suche über Hash-Algorithmus, einfacher Änderungsdienst (keinerlei Blockoperationen)	kein geordnetes Blättern

geordnetes Blättern: ausgehend von einem bestimmten Datensatz kann (alphabetisch/numerisch) vor- bzw. zurückgeblättert werden

Blockoperationen: Verschieben ganzer Datenblöcke, zum Beispiel zum Eintragen eines neuen Datensatzes

◄ Tabelle 2. Vor- und Nachteile der wichtigsten Datenstrukturen

Dateiform	Vorteile	Nachteile	benötigte Zugriffsart
sequentielle Datei	einfache programmtechnische Realisation	Dateigröße von Rechnerspeicherkapazität abhängig	sequentieller Zugriff
index-sequentielle Datei	Dateigröße nur von der Kapazität des externen Speichers abhängig (Floppy, Kassette)	schnelle Suche nur über den Index	direkter Zugriff
Hashing-Datei	Dateigröße nur von der Kapazität des externen Speichers abhängig, extrem schnelle Suche über Hash-Feld	schnelle Suche nur über Hash-Feld, Beschränkung bei der Wahl der Datenstruktur auf Hash-Organisation	direkter Zugriff

▲ Tabelle 3. Die wichtigsten Dateiformen

Handicap für Kassettenrecorder

Kurz erwähnt werden sollte noch die Art des Zugriffs auf Daten. Man unterscheidet hauptsächlich den direkten und den sequentiellen Zugriff (Tabelle 3). Mit einem Kassettenrecorder geht es nur sequentiell: Um einen bestimmten Datensatz zu finden, müssen alle in der Datei vorhergehenden Elemente gelesen werden. Damit ist weder die indexsequentielle Datei, bei der über einen Zeiger direkt auf den gewünschten Datensatz zugegriffen werden kann, noch die Hashing-Datei, die ebenfalls direkten Zugriff voraussetzt, möglich.

Bei der Floppy gibt es diese Einschränkungen nicht. Sie hat nur zwei Grenzen; zum einen die Speicherkapazität und zum anderen die Geschwindigkeit, mit der Daten übertragen werden. Und diese beiden Punkte sind es auch, die die Verwendung von Datenbanken auf dem C 64 fraglich machen. Die Verwendung mehrerer Dateien ist sehr speicherintensiv, und die Verknüpfung dieser Dateien erfordert einen häufigen Zugriff auf die Diskette und ist daher auch ein Zeitproblem. Interessant dürften Datenbanken eher beim neuen PC 128 werden. Warten wir's ab.

(Said Baloui/gk)

Sprachen für Computer (2)

Diesmal sollen uns bei unserem Überblick über die Welt der Programmiersprachen die modernen Sprachen interessieren. Sie erfahren, was Ada kann und was man sich unter Modula oder Lisp vorzustellen hat.

Die »klassischen« alten Programmiersprachen wie Fortran, Cobol oder Algol haben viel mehr miteinander gemeinsam, als man beim oberflächlichen Betrachten von Programmen in einer dieser Sprachen erwarten würde. So können zum Beispiel nur numerische Datentypen verwendet werden, eine Einschränkung, in der sich die unzulänglichen Hardwaremöglichkeiten der fünfziger Jahre widerspiegeln. Tatsächlich war Rechenzeit und Speicherplatz damals Mangelware, und niemand dachte auch nur im entferntesten daran, die teure Rechnerkapazität für andere Dinge als schwierige numerische Berechnungen einzusetzen. Das führte dann später in den sechziger und siebziger Jahren zur Unsitte, rein numerisch orientierte Sprachen wie Fortran schließlich für alle möglichen Zwecke einzusetzen. Die Sprache war halt einmal vorhanden, Rechenkapazität verfügbar und so wurden sogar ganze Textverarbeitungsprogramme in Fortran geschrieben, Buchstabe für Buchstabe fein säuberlich als Zahlenwert codiert und in einem numerischen Datenfeld abgelegt.

Da war eine Sprache wie Basic, das 1965 am Dartmouth College entwickelt worden ist, schon ein gewisser Fortschritt. Basic bietet neben numerischen Datentypen auch noch Strings, also Zeichenketten, als Datentyp an und stellt auch spezielle Operatoren dafür zur Verfügung. Tatsächlich ist mit der Möglichkeit, numerische Daten und Texte zu bearbeiten, schon ein großer Teil der (damals) denkbaren Computer-Anwendungen abgedeckt. Moderne Programmiersprachen stellen aber noch weitaus mehr Komfort zur Verfügung. Es können beliebige eigene Datentypen definiert werden, es gibt sogenannte »Verbundvariable«, zusammengesetzt aus mehreren Variablen verschiedener Datentypen, und schließlich können auch komplexe Datenstrukturen wie verkettete Listen oder Suchbäume einfach dargestellt werden. Die erste Sprache, die diese Möglichkeiten konsequent verwirklichte, war Pascal, auf das schon im ersten Teil ausführlicher eingegangen wurde. Im folgenden wollen wir uns etwas näher mit den Nachfolgern von Pascal beschäftigen, aber auch auf völlig andersartige Konzepte eingehen.

Pascal in Kurzform: C

Die Entwicklung der Sprache C ist eng verbunden mit der Geschichte des 16-Bit-Betriebssystems Unix, das selbst vollständig in C geschrieben ist. Der Name dieser Sprache hat eine verblüffend einfache Herkunft. Zur Programmier-Unterstützung auf Minicomputern von Digital Equipment wurden um 1970 Spezialsprachen entwickelt, die einfach nach dem Alphabet die Bezeichnungen A und B erhielten. Die B-Sprache wurde 1971 dazu benutzt, Unix, das damals noch in Assembler geschrieben war, auf einfache Art und Weise auf andere Computer zu übertragen. Schließlich erkannte Dennis Ritchie, ein Programmierer bei Bell Labs, die Fähigkeiten der Sprache B, erweiterte und verfeinerte sie und nannte das Resultat C.

Kurz darauf (1973) wurde Unix auf C umgeschrieben — das erste Betriebssystem, das in einer höheren Programmiersprache abgefaßt war. Das war nur möglich, weil C ein strukturiertes Sprachkonzept bei gleichzeitiger größtmöglicher Effizienz der Programme zur Verfügung stellt. In C geschriebene Programme sind einerseits sehr kompakt (geringer Speicherbedarf), andererseits sehr schnell (typischerweise etwa 50 mal so schnell wie Basic).

Programmierung in C besteht im wesentlichen im Schreiben von Funktionen, die jeweils Teilbereiche des Problems lösen. Die Gesamtlösung ergibt sich dann durch zweckmäßigen Einsatz dieser Funktionen. Dementsprechend ist C einerseits stark an Pascal angelehnt, verfügt andererseits aber über zusätzliche Befehle und Operatoren, die der verbesserten Ausnutzung der vorhandenen Hardware (= des Prozessors) dienen. So bietet C spezielle, an die Assembler-Programmierung angelehnte Befehle zum Inkrementieren und Dekrementieren (erhöhen/erniedrigen) von Variablenwerten und als einzige der verbreiteten höheren Programmiersprachen die Option, Variable als Register-Variable zu deklarieren. Das bedeutet, daß solche Variablen nach Möglichkeiten in internen Prozessor-Registern gehalten werden, was einen wesentlich schnelleren Zugriff als bei den normalerweise verwendeten Speichervariablen ermöglicht. An dieser Stelle merken Sie wahrscheinlich schon, daß bei der Entwicklung der Sprache nicht unbedingt an den 6502-Prozessor gedacht wurde, der mit seinen bescheidenen drei Registern (A,X,Y) in dieser Beziehung nicht gerade überwältigende Möglichkeiten bietet. Allerdings würde sich als Register-Ersatz die Zero-Page anbieten, bei der der Zugriff ja auch etwas schneller ist. Andererseits wurde C auch nicht gerade für 64-Bit-Superprozessoren konzipiert. Derartige in Großrechnern verwendete Prozessoren verfügen in der Regel über eine sehr große Zahl von Registern, und ein guter Fortran-Compiler für so einen Großrechner verwendet natürlich auch diese Register für die Programmoptimierung. So wird C in der Regel auf 16- oder 32-Bit-Maschinen eingesetzt. Bei den 8-Bit-Prozessoren ist C praktisch nur für den Z80 verbreitet (unter CP/M), allerdings soll eine Version für den C 64 in Vorbereitung sein. Wir werden Sie im 64'er Magazin darüber auf dem laufenden halten.

Modulare Programmierung: Modula

Das Grundkonzept aller modernen Programmiersprachen heißt Modularisierung der Programmentwicklung. Damit ist gemeint, Probleme nicht mit immensem Aufwand in umfangreichen Programmen zu lösen (die ja für jedes Problem wieder

neu geschrieben werden müssen), sondern Teilaspekte des Problems in möglichst allgemeiner Form in eigenständigen Teilprogrammen, eben den sogenannten Modulen, zu lösen. Der Vorteil dieser Art der Software-Entwicklung liegt auf der Hand: Ein Teilproblem muß nur ein einziges Mal gelöst werden; tritt das gleiche Teilproblem irgendwann einmal wieder bei der Programmierung auf, kann man auf das fertige Modul zurückgreifen. Außerdem ist es natürlich zumeist viel einfacher, mehrere kleinere Probleme zu lösen als ein großes. Bei konsequentem Einsatz der »modularisierten Programmierung« lassen sich Programme somit auf lange Sicht wesentlich rationeller entwickeln.

Allerdings gibt es einige Voraussetzungen für die Modularisierung: Die einzelnen Module müssen über genau definierte »Schnittstellen« nach »außen« verfügen, damit sie zweckmäßig ausgewählt und eingesetzt werden können. Andererseits darf aber das »Innenleben« der Module auf keinen Fall irgendwelche Dinge »außen« beeinflussen. Basic ist daher zum Beispiel sehr schlecht für diese Art der Programmierung geeignet: Unterprogramme haben keine Namen, sondern werden über Zeilennummern aufgerufen (schlechte Dokumentation), es gibt keine Parameterübergabe, also keine feste Schnittstelle, und schließlich ist jede Variable, die im Unterprogramm verändert wird, anschließend auch im Hauptprogramm verändert.

Modula, entwickelt von Professor N. Wirth, dem Schöpfer der Sprache Pascal, ist eine der ersten Sprachen, die das Konzept der Modularisierung, das zum Teil ja auch schon in Pascal enthalten ist, konsequent durchführt.

Jedes Modula-Programm ist selbst ein Modul und kann dementsprechend andere Programme als Untermodule verwenden. Dabei gibt es keinen Unterschied zwischen den Standard-Modulen, die die Sprache dem Benutzer zur Verfügung stellt und selbstgeschriebenen (Programm-)Modulen. Jedes Modul ist in sich vollständig abgeschlossen, daß heißt Vereinbarungen über Datentypen, Funktionen und Variable gelten nur innerhalb des Moduls. Sollen die in einem Modul getroffenen Definitionen anderen Modulen zugänglich gemacht werden, dann müssen sie vom Modul »exportiert«, von den anderen Modulen »importiert« werden. Damit werden alle unerwünschten Nebeneffekte, wie sie etwa bei Basic-Unterprogrammen häufig auftreten können, vollständig ausgeschlossen.

Selbstverständlich sind die einzelnen Module intern ebenfalls völlig strukturiert aufgebaut. Den Befehl GOTO sollte man als Modula-Programmierer schnellstens vergessen; dafür stehen die von Pascal gewohnten Schleifenstrukturen (FOR...TO... DO...END, REPEAT...UNTIL, WHILE...END) zur Verfügung, zusätzlich noch die LOOP...EXIT...END-Schleife, die alle Befehle zwischen LOOP und END so lange durchläuft, bis die bei EXIT angegebene Bedingung erfüllt ist. Diese Schleifenstrukturen sind übrigens ohne Ausnahme auch im Commodore 3.5-Basic des C 16, oder im 7.0-Basic des C 128 vorhanden, ein sicheres Zeichen, daß sich das Konzept der strukturierten Programmierung auch im Bereich der Basic-Heimcomputer immer mehr durchsetzt. Leider ist Modula noch nicht für Commodore-Computer erhältlich.

Die Krönung: Ada

Den bisherigen Höhepunkt bei der Entwicklung moderner, modularer Programmiersprachen stellt ohne Zweifel die Sprache Ada dar.

Die Sprache hat ihren Namen nach der Countess Augasta Ada Lovelace, die sich als erste Frau bereits im vorigen Jahrhundert (!) mit Algorithmen und Rechenmaschinen beschäftigte. Bekannt wurde sie in erster Linie, indem sie ein »Rechenkalkül« für die von Charles Babbage entworfene mechanische Rechenmaschine entwarf. Dieses Kalkül kann man im weitesten Sinne als den ersten Algorithmus für eine Maschine bezeichnen.

Ada wurde erst in jüngster Zeit im Auftrag des amerikanischen Verteidigungsministeriums entwickelt und baut ebenso wie C und Modula auf dem Konzept von Pascal auf. Die Sprache ist ebenso wie Modula streng modular und strukturiert aufgebaut, geht aber von den Fähigkeiten weit über Modula hinaus.

Wie in Modula gehören praktisch alle Funktionen, insbesondere auch die Ein-/Ausgabe-Operationen nicht direkt zur Sprache, sondern werden in besonderen Modulen, den sogenannten Packages (Pakete) bereitgehalten. Das hat den Vorteil, daß der Anwender im Bedarfsfalle fast alle Funktionen selber neu schreiben kann, wenn ihm die Standard-Funktionen nicht gefallen. Alle Packages, ob selbstgeschrieben oder vom System bereitgestellt, müssen ins Programm, das seinerseit ein Package darstellt, importiert werden. Dies geschieht sehr einfach durch Angabe des entsprechenden Paket-Namens. Um zum Beispiel die Standard-Routinen für die Ein- und Ausgabe von Texten zu importieren, verwendet man die Anweisung:

```
TEXT__IO IS PACKAGE
STANDARD__TEXT__IO
```

Durch diese Angabe werden die im Modul STANDARD__TEXT__IO enthaltenen Funktionen dem Programm-Modul unter dem Namen TEXT__IO bekanntgemacht. Das Standard-Modul enthält beispielsweise die Funktionen WRITE und READ zum Drucken und Einlesen von Daten.

Durch die Anweisung USE TEXT__IO werden dann bei Bedarf (wenn WRITE oder READ im Programm vorkommt) die entsprechenden Funktionen aus dem STANDARD__TEXT__IO-Modul (das ja innerhalb des Programms den Namen TEXT__IO erhalten hat) entnommen und ins Programm eingefügt. Will man aber an einer bestimmten Stelle nicht die Standardfunktion verwenden, sondern eine selbstgeschriebene, dann braucht man das eigene Modul nur mit

```
MEIN__TEXT IS PACKAGE
EIGENE__TEXT__IO
```

anzumelden und im Programm mit

```
USE MEIN__TEXT
```

auf die selbstgeschriebenen Routinen umzuschalten. Dabei wird die Ada-Fähigkeit des »Überladens« von Funktionen verwendet, was nichts anderes bedeutet, als daß ein- und dasselbe Schlüsselwort (oder Zeichen) je nach Kontext grundverschiedene Funktionen bezeichnet. Nach Umschaltung mittels »USE« heißt die Ausgabefunktion in unserem Beispiel zwar immer noch »WRITE«, es wird aber die selbstgeschriebene Funktion aufgerufen und nicht die Standard-Funktion.

Natürlich kann man diese Technik auf alle möglichen Bereiche ausdehnen. Damit wird zusammen mit der Möglichkeit sich eigene Datentypen, wie in Pascal oder Modula, zu definieren, ein sehr hoher Grad an Flexibilität gewährleistet. Zum Beispiel kann man sich bei Bedarf einen Datentyp COMPLEX für komplexe Zahlen definieren und die Operatoren »+«, »-«, »*« und »/« durch Überladen auch auf diesen neuen Datentyp anwenden.

Zwei weitere Aspekte von Ada sind noch besonders erwähnens-

wert: Zum einen existieren sehr komfortable Möglichkeiten, Ausnahmesituationen im Programm (sogenannte Exceptions) ohne Abbruch unter Kontrolle zu bringen. In der Regel dient das zum Abfangen von Fehlerbedingungen, die innerhalb eines Programms auftauchen können (in etwa vergleichbar mit »ON ERROR« oder »TRAP« in Basic). Eine solche Fehlerbehandlung ist bei Compilersprachen bislang sehr selten, für die Zwecke des amerikanischen Verteidigungsministeriums aber natürlich unabdingbar. Man stelle sich nur vor, in kritischen Situationen würden Radarstationen keine Informationen mehr weitergeben wegen eines »DIVISION BY ZERO ERROR« ...

Der zweite zusätzliche Aspekt von Ada ist die Fähigkeit, parallele Prozesse automatisch durchführen zu können (Multitasking). Damit ist gemeint, daß bestimmte Operationen gleichzeitig ablaufen können (sofern der Computer mit mehreren Prozessoren ausgestattet ist, sonst ergibt sich natürlich nur eine »Pseudo-Gleichzeitigkeit«). Das bringt natürlich eine unter Umständen immense Erhöhung der Verarbeitungsgeschwindigkeit, allerdings auf Kosten eines hohen Hardwareaufwandes.

Wer sich etwas näher mit Ada beschäftigen will und einen C 64 besitzt, der findet im Ada-Trainingskurs von Data Becker (siehe Test in dieser Ausgabe) einen brauchbaren Einstieg.

Es gibt ganz grob eingeteilt zwei grundverschiedene Linien unter den Programmiersprachen.

Da sind einmal die sogenannten Anweisungs-Sprachen, zum anderen die funktionalen Sprachen. Zu den Anweisungs-Srachen gehören alle »klassischen« Programmiersprachen einschließlich Basic, daneben natürlich Pascal, Modula, Ada und C. Alle diesen Sprachen sind von Ihrer Struktur her mehr oder weniger stark an die Hardware-Organisation der heutigen Computer angelehnt (Befehl holen — Befehl ausführen — Programmzeiger auf nächsten Befehl setzen). Funktionale Sprachen verlangen demgegenüber nicht die Ausführung spezieller, einzelner Befehle, sondern beschreiben Strukturen und bilden neue Strukturen.

Vertreter dieser Sprachen sind zum Beispiel Lisp (eine listenverarbeitende Sprache), Snobol (eine symbolische Simulations-Sprache) und eine Unzahl kleiner, außerhalb der Universitäten völlig unbekannten Spezialsprachen.

Lisp ist sicherlich der bekannteste Vertreter dieser Sprachenklasse. Für den Laien besteht Lisp in erster Linie aus einer Unzahl von öffnenden und schließenden Klammern, ab und zu findet sich dazwischen auch mal ein anderes Zeichen. Die Ursache dafür ist die funktionale Struktur der Sprache. Es gibt in Lisp keine Befehle, sondern nur Funktionen, die aus primitivsten Grundelementen sehr komplexe Strukturen aufbauen können. Ausgangspunkt ist immer die kleinste, unteilbare Lisp-Struktur, das Atom. Ein Atom kann eigentlich alles sein, was man über eine Tastatur in einen Computer hineinbekommt. Mit (ATOM APFEL) wird ein »DING« namens Apfel definiert, daß weder interne Struktur noch Beziehungen zu irgendwelchen anderen Daten hat. Diese Beziehungen zu anderen Daten werden erst im folgenden durch Funktionen definiert. Wesentliches Element von Lisp (List Processing Language) ist die Fähigkeit, Listen zu bilden. Eine Liste ist eine geordnete Aufzählung von Elementen. Diese Elemente sind im einfachsten Fall Atome, es kann sich dabei aber auch um andere Listen oder gar um Funktionen handeln. Mit

(SETQ OBST (APFEL BIRNE KIRSCHE))

wird eine Liste OBST definiert, die aus den Atomen Apfel, Birne und Kirsche besteht (die natürlich vorher als Atome vereinbart worden sind).

Funktionale Sprachen

Es gibt jetzt eine Menge vordefinierter Funktionen, um diese Listen weiter zu verarbeiten. Es können einzelne Elemente oder ganze Listen zu neuen Listen umgewandelt werden, es können Bedingungen festgelegt werden, unter denen das geschieht und so fort. Dabei können sehr komplexe Strukturen, regelrechte logische Netzwerke, entstehen.

Dadurch wird in gewisser Weise der in erster Linie assoziativ arbeitende Mechanismus des menschlichen Denkens viel besser simuliert, als durch schrittweise Abarbeitung von Befehlen.

Lisp wird denn auch mit durchaus beachtenswerten Erfolgen bei praktisch allen KI-(künstliche Intelligenz-)Projekten eingesetzt. Die Sprache liefert vor allem auf den Gebieten der symbolischen Datenverarbeitung (nicht-numerische Mathematik), der Erkennung von Mustern und dem logischen Folgern gute Ergebnisse.

Für den C 64 ist Lisp nicht erhältlich, jedoch ist mit Logo eine andere Sprache verfügbar, die ebenfalls Listenverarbeitung unterstützt.

Dieser Aspekt der Listenverarbeitung ist allerdings bei Logo längst nicht so konsequent implementiert wie in Lisp, aber dafür auch für den Einsteiger recht schnell zu durchschauen. Bekannt geworden ist Logo allerdings in erster Linie durch die »Turtle-Grafik«, die schon zum Markenzeichen dieser Sprache geworden ist.

Die Turtle (Schildkröte) wird in der Regel durch ein kleines Dreieck am Bildschirm dargestellt, das mit Befehlen wie FORWARD, BACK, LEFT und RIGHT in alle Richtungen bewegt werden kann. Dabei hinterläßt die Schildkröte eine sichtbare Linie auf dem Bildschirm und kann so zum bequemen Zeichnen auch von komplexen Grafiken eingesetzt werden.

Wie bei Basic, so handelt es sich auch bei Logo (übrigens auch bei Lisp) um einen Interpreter, was ein sehr bequemes Vorgehen beim Programmieren erlaubt. Alle Programmbefehle können auch im Direktmodus eingesetzt werden, und so kann man alle Routinen direkt im Dialog mit dem Computer austesten, ohne das gesamte Programm wie bei einem Compiler nach jeder Änderung ständig wieder neu übersetzen zu müssen.

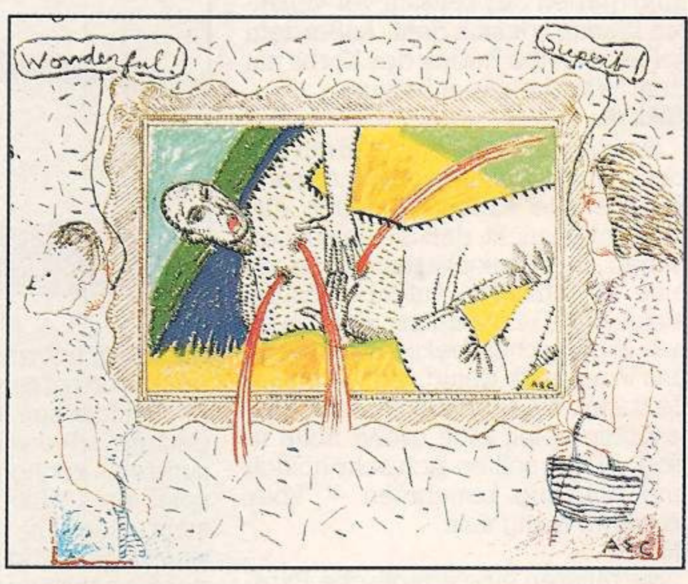
Die einfache Handhabung der Grafik ist sicherlich einer der Hauptgründe, die speziell beim C 64 für die Verwendung von Logo sprechen. Wenn Sie sich für Logo interessieren, dürfen wir Ihnen unseren Testbericht über das Commodore-Logo in dieser Ausgabe empfehlen.

Natürlich ist es mit den hier behandelten Sprachen noch nicht getan. Es existiert eine Unmenge weiterer Programmiersprachen, mit deren Behandlung man ganze Bücher füllen könnte. Dieser Bericht sollte Ihnen aber einen allgemeinen Überblick gegeben haben, wo's bei den Programmiersprachen lang geht.

Speziell für den C 64 wird es sicher in naher Zukunft eine Reihe weiterer Sprachen geben. Interessant wird die Angelegenheit natürlich auch im Hinblick auf den neuen Commodore 128 PC, der ja die ganze Welt der Programmiersprachen unter CP/M zugänglich macht. Ganz sicher werden wir das Thema »Programmiersprachen« unter diesem Aspekt nochmals aufgreifen. (ev)

Infocom-Geheimnisse gelüftet?

Bekannt dürften sie mittlerweile sein, die Adventurespiele von Infocom. Gelöst sind sie noch lange nicht. Ein Grund, Ihnen Tips dazu zu geben.



Beispiel einer Illustration zum Infocom-Adventure »Suspect«. In diesem Kriminal-Adventure gilt ausnahmsweise nicht der Spruch »Der Mörder ist immer der Gärtner«.



Die Infocom-Adventures sind wohl die besten Text-Abenteuerspiele, die es gibt. Sie verstehen in der Regel einen Wortschatz von über 800 Wörtern, akzeptieren ganze Sätze und lesen sich wie spannende Romane — kein Wunder, denn Infocom läßt auch Romanautoren für sich arbeiten. Was für die einen der Unterschied zwischen Bilderbuch und ausgereiftem Roman, ist für die Adventurefans die Unterscheidung zwischen Grafik- und Textabenteuer. Die Textadventures verbrauchen keinen Speicherplatz für Grafiken und bieten deswegen viel mehr andere Möglichkeiten. Dieser für die anderen Dinge freie Speicherplatz wird meist ausgenutzt, um einen enormen Umfang an Räumen, Gegenständen und Aufgaben unterzubringen. Und da die Infocom-Adventures genau dies tun, sind sie sehr schwer lösbar. Daher unsere Tips.

Tips für alle Infocoms

Als erstes erhalten Sie ein paar allgemeine Tips, die Sie eigentlich für alle Infocom-Adventures benutzen können.

Der Parser ist, einfach gesehen, das Stück des Programms, das die Eingabe der Sprache steuert. Der Infocom-Parser ist zwar einer der besten seiner Art; aber was immer auch so toll daran sein mag — es ist zu empfehlen, den Infocom-Parser einfach wie bei einem Zwei-Wort-Adventure anzusehen. Das spart eine Menge Zeit. Mit dem Wortschatz herumprobieren sollten Sie trotzdem: In jedem Infocom-Abenteuer gibt es Worte, die die anderen Adventures aus diesem Softwarehaus nicht haben, und die manchmal zum Ziel führen können.

Und noch etwas: Denken Sie nie daran, was Sie machen würden, wenn Sie tatsächlich so ein Abenteuer durchleben müßten. Überlegen Sie sich, was der Autor an Gemeinheiten eingebaut haben könnte — das führt Sie viel eher zum Ziel.

Außerdem gibt es sehr selten zwei Methoden, ein Problem zu lösen. Speichern Sie also immer Ihr Spiel ab, bevor Sie mit etwas herumexperimentieren. Beispiel: Wenn Sie einen Gegenstand an einem Ort falsch benutzen und später einen Ort erreichen, an dem Sie den Gegenstand (oder Zauberspruch) unbedingt benötigen, müssen Sie den abgespeicherten Spielstand wieder laden (weil der Gegenstand oder Zauberspruch abgenutzt oder verbraucht ist). Es sei denn, Sie wollen von vorn beginnen.

Gegenstände:

Untersuchen Sie alle Gegenstände sehr genau. Viele Dinge können manipuliert und verändert werden, andere haben ein »Innenleben«, das heißt sie können Flüssigkeiten oder andere Gegenstände beinhalten. Ein Eigenleben der Objekte ist auch nie auszuschließen; achten Sie also auf alles, was eigenständig sein könnte. Das Schwert aus der Zork-Serie leuchtet beispielsweise blau, wenn Gefahr nahe ist, und rot, wenn der Träger des Schwertes verärgert ist. Legen Sie die Gegenstände immer ab, wenn Sie sie einmal sinnvoll benutzt haben. In Infocom-Adventures werden sie meistens kein zweites Mal benötigt. Wenn Sie einen Gegenstand mehrmals untersuchen, und keine nähere Beschreibung folgt, können Sie den Gegenstand ruhig als unwichtig betrachten und vergessen. Letzteres gilt für alle Infocom-Adventures außer »Hitchhikers Guide to the Galaxy«, das extrem von der Firmennorm abweicht.

Personen:

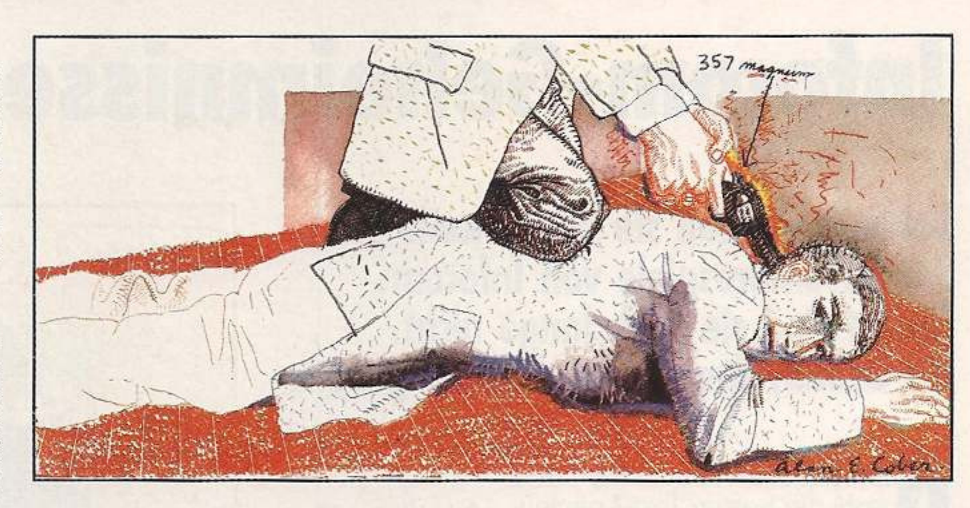
Um zu jemandem zu sprechen, müssen Sie als erstes den Namen der Person eingeben. Um beispielsweise den Dieb zu fragen, ob er Ihnen das Stilett geben würde, müssen Sie tippen: »Thief, give me the stiletto«.

Fragen Sie die Leute nach Gegenständen, anderen Personen und speziellen Ereignissen. Untersuchen Sie auch die Personen genau (mit EXAMINE).

Soweit zu allgemeinen Tips. Zu den für die einzelnen Abenteuer angegebenen Hinweisen ist noch zu sagen, daß wir hier nur kleine Denkstützen geben wollen. Wer mehr erfahren will, kann sich die in Geheimtinte gedruckten »Hint Books« von Infocom selbst bestellen. Wir für unseren Teil arbeiteten völlig ohne Lösungsbücher, befragten dafür aber viele Abenteurer, aus deren Erfahrungen sich die folgenden Tips zu den speziellen Infocom-Adventures ergeben.

ZORK I

Zork ist ein Spiel, bei dem es darum geht, zwanzig Schätze zu finden und sie in einer Schatzkiste abzulegen. Ein Dieb ist das Haupthindernis dabei, weil er versucht, Sie auszurauben oder umzubringen. Er kann zwar getötet werden, aber nur, wenn Sie stark genug sind (was von der Anzahl der Schätze in der Schatzkiste abhängt) und wenn Sie die richtige Waffe besitzen. Diese Waffe ist scharf, aber nicht das Schwert! Sobald Sie einen Schatz finden, sollten Sie Richtung Haus gehen.



ZORK II

Hier gibt es zehn Schätze, aber das Hauptziel ist es, den Zauberer von »Frobozz« zu vernichten. Es ist sehr wichtig, herauszufinden, was im Irrgarten der seltsam verwinkelten Räume vor sich geht. Außerdem sollten Sie den Zweck des Korbs mit dem Wäschebeutel kennen.

ZORK III

Sie können in diesem Spiel nur sieben Punkte bekommen. Und wenn Sie die alle haben, bedeutet das noch lange nicht, daß Sie gewonnen haben. Die Punkteverteilung ist auf eine »potentielle« Aufgabenlösung ausgelegt. Das bedeutet, daß Sie einen Punkt schon bekommen, wenn das Programm denkt, Sie wären nahe daran, ein Problem zu lösen. Philosophie spielt eine große Rolle in Zork III. Sie sollten sich sehr höflich und mildtätig benehmen — aber trotzdem mutig sein.

INFIDEL

Sie müssen eine ägyptische Pyramide erforschen, in der Hieroglyphen an die Wand geschrieben sind. Lernen Sie die Hieroglyphen gut, denn in ihnen steht die Lösung aller Probleme. Entziffert ergeben sie übrigens reines Englisch. Tip: Viele Objekte haben ein einzelnes Hieroglyphen-Zeichen, das für sie steht. Nehmen Sie in der Pyramide alles mit, was Sie finden können, besonders den Mast. Übrigens: Die Pyramide steht fast neben dem Camp.

STARCROSS

In diesem Spiel wurde viel Wert auf die Manipulation von Gegenständen gelegt. Viele Gegenstände sind farbig, um die Unterscheidung für den Spieler einfacher zu machen. Karten zu zeichnen ist hier wesentlich einfacher als bei den Zorks. Die farbigen Stäbe dienen als »Schätze« und müssen in die richtigen Slots gesteckt werden. Ein bißchen Wissen aus der Chemie (Atomhülle, Elektronen etc.) wird Ihnen helfen. Bevor Sie das Schiff verlassen, sollten Sie die vier farbigen »docking ports« des Alien-Raumschiffs genau untersuchen. Geben Sie Ihren Raumanzug dem Chef, und folgen Sie ihm durch den »maze« zu einem anderen docking ports.

PLANETFALL

Dieses Spiel basiert mehr auf Komplexität als auf schwierigen Problemen. Floyd **muß** sterben, bevor Sie gewinnen können. Betrachten Sie genau das Loch in der Wand und das verdächtige Gemälde, das in einem der Büros hängt.

SUSPENDED

Sie kontrollieren sechs Roboter, die Ihnen Informationen senden. Jeder davon hat seine eigenen speziellen Fähigkeiten, die, richtig verstanden und angewandt, des Rätsels Lösung sein können. Iris kann zum Beispiel ein TV-Bild von dem, was sie sieht, senden. Konzentrieren Sie sich also ständig auf Iris. Die Farben der Gegenstände helfen Ihnen, herauszufinden wie man sie benutzt. Steuern Sie die Roboter so, daß die Menschen in unvermeidbare Fallen laufen.

ENCHANTER

Enchanter läßt sich nur mit Hilfe von Zaubersprüchen lösen. Die »ewige Treppe« muß zerstört werden. Sie benötigen auch unbedingt die Hilfe des Abenteurers, der Ihnen begegnet. An einer anderen Stelle sollten Sie Bleistift und Radiergummi benutzen. Genaueres zu Enchanter finden Sie im 64'er, Ausgabe 3/85.

SORCERER

Noch mehr Magie — und das schwerste aller Infocom-Adventures. »Beleuchten« Sie sich selbst (Zauberspruch »Frotz«), und gehen Sie dann los, um Belboz zu suchen. Benützen Sie den Infotater, um die Schatzkiste zu öffnen. Manche Zaubersprüche müssen zweimal gelernt werden, um die gewünschten Resultate zu erzielen. Speichern Sie nach jedem gelösten Problem ab, denn Sorcerer ist sehr schwer. Sie benötigen unbedingt die orange Flüssigkeit, bevor Sie den »Yonk«-Spell an einer bestimmten Stelle benutzen. Heben Sie sich den »Vezza«-Spell bis ganz zum Schluß auf.

DEADLINE und WITNESS

In beiden Spielen wird kriminalistischer Spürsinn verlangt. Hauptarbeit ist es, die Leute über andere Personen und bestimmte Situationen auszufragen. Folgen Sie den Leuten unauffällig.

In Deadline müssen Sie »George« einen bestimmten Gegenstand zeigen, und zwar vor der Testamentsverlesung. Es gibt außerdem einen Geheimgang und einen versteckten Safe.

In Witness sollten Sie sich im Arbeitszimmer aufhalten, wenn »Monica« eintritt. Täter ist (wie immer) derjenige, der am wenigsten danach aussieht.

HITCHHIKER'S GUIDE TO THE GALAXY

Dieses Spiel weicht stark von der »Infocom-Norm« ab, weil es sehr gemein und heimtückisch gemacht wurde.

Legen Sie sich vor dem Bulldozer in den Dreck und warten Sie. Nehmen Sie nicht das Handtuch, das Ihnen angeboten wird. Später im Adventure sollten Sie Ihre Sinne genau nachzählen; benutzen Sie den, der nicht auf dem Bildschirm steht. Manchmal werden Sie vom Programm angelogen. Versuchen Sie also alles mehrmals.

SUSPECT

Dieses Spiel läuft ähnlich wie Deadline und Witness ab. Allerdings sind Sie diesmal der Mordverdächtige und müssen sich innerhalb ein paar Stunden vom Verdacht befreien. Achten Sie sehr genau auf Gespräche anderer Leute (auch Wortfetzen fügen sich mitunter zu wichtigen Indizien zusammen). Fragen Sie jeden über alles aus, was Ihnen einfällt. Besonders viel erzählt der Barkeeper, dem Sie sich als erstes zuwenden sollten.

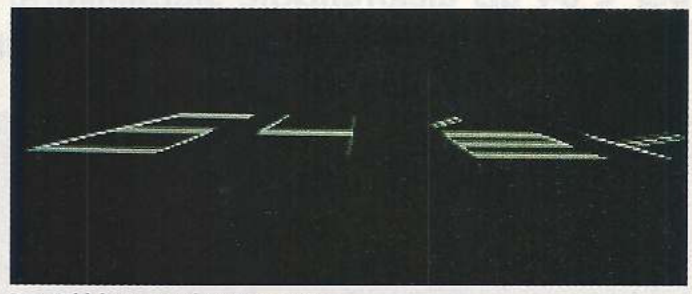
CUTTHROATS

Dieses Adventure ist eines der neuesten aus der Infocom-Reihe. Daher läßt sich hierzu noch nicht viel sagen. Wenn Sie einen Tip dazu haben — immer her damit!

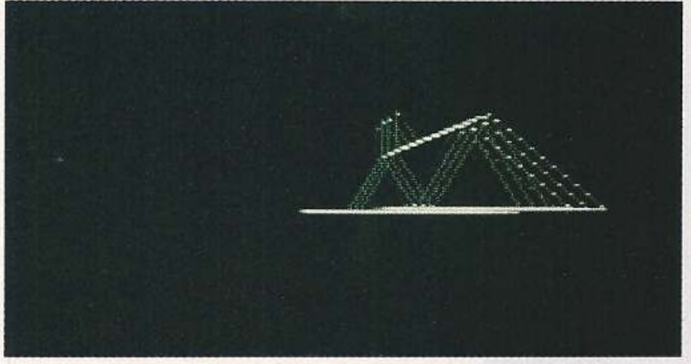
Die oben genannten Hinweise zu den Adventures sind, wie gesagt nur minimale Denkanstöße für den Abenteuerspieler. Wer mehr wissen will, sollte sich die »Hint Booklets« von Infocom aus USA bestellen. Die Lieferzeiten sind allerdings sehr lang. (M. Kohlen/rg)

Trickfilm mit dem C 64

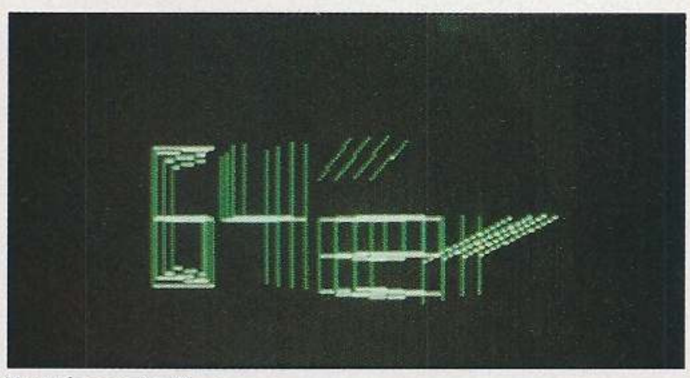
In die vierte Dimension, die bewegte dreidimensionale Grafik, dringen Sie mit diesem Programm vor. Sie können so mit einfachsten Mitteln Trickfilme mit verblüffenden Eigenschaften erstellen.



Das Bild kann »gekippt« werden



Auch dreidimensionale Körper können dargestellt werden



Vier Phasen der Bewegung

Die Idee zu diesem Programm kam uns, als wir zum zweiten Mal die Sendung »Das Bild, das aus dem Rechner kam« sahen.

Die erste noch in Comal geschriebene Version berechnete jeweils ein Bild und zeichnete es sofort. Das Ergebnis: nur alle 7 Sekunden ein Bild! Auch als erst alle Linien berechnet und dann gezeichnet wurden, konnte von einem flüssigen Bewegungsablauf keine Rede sein. Bei der jetzigen Version wird der Teil des Zeichnens von einer Maschinencode-Routine übernommen, die eine Geschwindigkeit von 3 bis 10 Bilder/s erlaubt. Die Unterschiede gegenüber anderen Trickfilmgeneratoren:

— Es muß nur ein Bild eingegeben werden, alle anderen Bilder berechnet der Computer aus diesem Bild.

— Auch komplizierte Bewegungsabläufe können mühelos erzeugt werden (zum Beispiel, um einen Körper um die Z- und X-Achse zu drehen, dabei heranzuholen und nach links zu bewegen, bis er den Bildschirm verläßt, braucht man nur eine DATA-Zeile!).

— Es können sehr viele (bis zu 255) und große Bilder gespeichert werden, da nur die Linienkoordinaten gespeichert werden.

— Es werden keine Basic-Erweiterungen benötigt.

(Dirk und Armin Biernaczyk/rg)





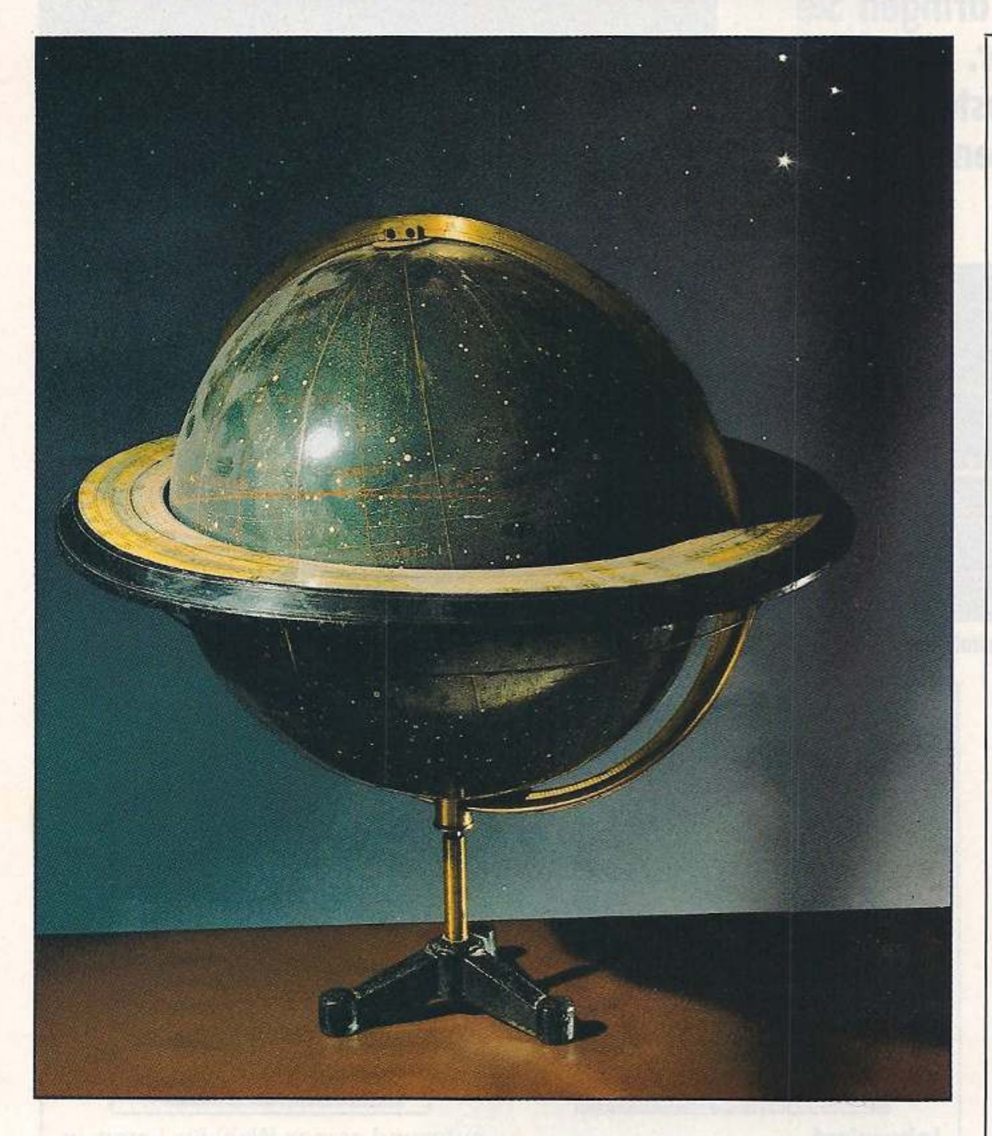
Lebenslauf

Wir erblickten am 14.4.1968 das Licht der Welt. Und nicht nur das, sondern jeder noch ein anderes Geschöpf: wir waren Zwillinge. Nachdem wir sechs Jahre zusammen mehr oder weniger ruhig verbracht hatten, wurden wir 1974 in die Grundschule West eingeschult. Dort überstanden wir die ersten 4 Jahre unseres Schullebens ohne Komplikationen. 1978 wechselten wir aufs Märkische Gymnasium, was keine große Umstellung bedeutete, denn auch hier reichte das Zeugnis immer aus, um die Geldbeutel der Verwandten zu öffnen. Meinen ersten Kontakt mit Computern hatte ich (Dirk) in der Schule in der 10. Klasse mit Logo, was meinem Bruder allerdings aufgrund seiner Wahl für Latein in der Klasse 9 nicht möglich war. Kurze Zeit darauf sammelten wir zusammen in der Volkshochschule Bochum unsere ersten Basic-Kenntnisse. Fasziniert von der Computerei kauften wir uns Ende 1983 einen C 64 mit Diskettenlaufwerk. Da wir nach Abschluß der 10. Klasse von der Schule immer noch nicht die Nase voll hatten, machten wir weiter und besuchen heute die 11. Klasse und versuchen dort im Informatikunterricht, mittlerweile zusammen und in Pascal, den Wurm im Apfel II zu dressieren. Zu Hause schlagen wir uns lieber mit Basic, Assembler und ein wenig Comal herum. Das Ergebnis unserer Programmierwut ist dieses Programm.

(Dirk und Armin Biernaczyk)

Weißt Du wieviel Sternlein stehen ...

Der C 64 als elektronische Sternenkarte bringt Klarheit ins Dunkel der Nacht. Das Programm zeigt Ihnen, wo Planeten und Sternbilder am Himmel stehen.



Das Programm entstand aus dem Bedürfnis, den Standort von Sternen, Sternbildern und Planeten zu ermitteln. Vor allem die Planeten lassen sich so ohne weiteres mit dem bloßen Auge, nicht erkennen. Man kann sie nur identifizieren, wenn man genaue Informationen über ihren jeweiligen Standort hat. Deshalb wurde die Berechnung der Planetenbahnen, der Bahn der Sonne und der des Mondes in das Programm aufgenommen. Das Programm erstellt eine Sternenkarte für einen eingegebenen Standort zum gewünschten Beobachtungszeitpunkt.

Die Sternenkarte enthält die hellsten und bekanntesten Fixsterne beziehungsweise Sternbilder des nördlichen Sternenhimmels.

Sehr interessant ist es, neben der Ausgabe der Sternenkarte für eigene Beobachtungen, sich die Sternenkarte für jeden Punkt der Erde zeichnen zu lassen.

So kann man sich zum Beispiel in Gedanken auf den Nordpol versetzen. Der Polarstern (Polaris) steht im Zenit, also genau über dem Beobachter. Gibt man als Beobachtungszeitpunkt den 21. März (Frühlingsanfang für die Nordhalbkugel) ein, kann man feststellen, daß die Sonne den ganzen Tag über scheint.

Am Nordpol hat der Tag begonnen. Die Sonne scheint nun ein halbes Jahr ohne Unterbrechung. Erst zum Herbstanfang geht sie wieder unter. Es ist dann für ein halbes Jahr Nacht. Auch die Verhältnisse am Äquator (Breite 0°) sind sehr interessant.

(H. Hinkelmann/hm)



Lebenslauf

Ich wurde am 02.10.26 in Ahlen (Westfalen) geboren und lebe seit 1951 in Hamm. Seit dem 01.04.43 bin ich in der Finanzverwaltung und dort seit 1955 als Betriebsprüfer tätig.

Bereits als Kind interessierte ich mich sehr für Naturwissenschaften und Technik. Ich baute mir selbst ein Fernrohr, um die Sterne beobachten zu können. Ende der 60er Jahre begann ich mit dem Basteln in der Mikroelektronik. Es entstand eine Digitaluhr mit Kalenderfunktion. Diese stellte sich automatisch nach dem Zeitzeichen des WDR. Mein erster »Computer« war der programmierbare Taschenrechner Casio FX 501 mit 112 (!) Programmschritten. Es folgte der Sharp CP 1211 und dann der VC 20. Meinen C 64 besitze ich seit etwa einem Jahr. Von Anfang an wurden die Computer bei den Betriebsprüfungen und hier insbesondere bei den Prüfungsvorbereitungen und den Berichtsabfassungen eingesetzt. Seit einiger Zeit bin ich Mitglied des Arbeitskreises EDV bei der Oberfinanzdirektion Münster und für die Einführung der Datenverarbeitung im Prüfdienst tätig.

Die ersten Steuerberechnungen nahm ich noch mit dem Taschenrechner vor. Es folgten dann Programme für den VC 20 und nun für den C 64. Inzwischen habe ich kaum noch Zeit, private Programme zu entwickeln.

(Horst Hinkelmann)


64ER ONLINE

Checksummer 64

Der Checksummer 64 überprüft jede Basic-Zeile direkt nach der Eingabe und erspart deshalb eine aufwendige Fehlersuche.

Der Checksummer 64 ist ein kleines Maschinenprogramm, das Sie sofort unterrichtet, ob Sie die jeweilige Programmzeile korrekt eingegeben haben.
So gehen Sie vor:

1. Programm abtippen und speichern.
2. starten mit RUN
3. nach kurzer Zeit sehen Sie am Bildschirm:
Checksummer 64, Checksummer aktiviert, ausschalten mit Poke 1,55, anschalten mit Poke 1,53, Ready.
4. Anschalten des Checksummer 64 mit Poke 1,53.
5. Test: Geben Sie in einer freien Zeile ein: »1 REM« und drücken die Return-Taste. Am Bildschirm oben links sollten Sie die Prüfsumme <144> sehen.
6. Geben Sie ein Listing aus unserem Heft ein. Nach jeder Zeile wird die Zahl, die im Listing in Klammern < > steht, in den Bildschirm eingeblendet. Stimmen die Zahlen nicht überein, so liegt vermutlich ein Eingabefehler vor. **Die Zahl in den Klammern und auch die Klammern selbst dürfen beim Abtippen nicht mit eingegeben werden!**
7. Achten Sie bitte darauf, Zahlen und Zeichen nicht zu vertauschen. So ergibt zum Beispiel die Zahl 210 in einer Basic-Zeile die gleiche Prüfsumme wie 201.
8. Unsere Basic-Listings enthalten keine Grafikzeichen mehr. Diese werden ersetzt durch Klartext und stehen zwischen geschweiften Klammern. Deshalb sind weder die Klammern noch was dazwischen steht, abzutippen, sondern die in der Tabelle aufgeführten Tasten zu drücken. Auf Ihrem Bildschirm erhalten Sie dann wieder die entsprechenden Grafikzeichen.
Eine ausführliche Beschreibung finden Sie in den Ausgaben 1 bis 4/85. Ebenso den Checksummer für den VC 20.

```
10 REM ******************************     <175>
20 REM *                            *     <247>
30 REM *       CHECKSUMMER 64       *     <162>
33 REM *                            *     <004>
36 REM *       (VERSION  2.0)       *     <014>
40 REM *                            *     <011>
50 REM *           64'ER            *     <061>
60 REM *                            *     <031>
70 REM *        COMMODORE 64        *     <056>
80 REM *                            *     <051>
90 REM ******************************     <255>
100 PRINT"{CLR,13SPACE,RVSON}CHECKSUMMER 6
    4{RVOFF}"                             <025>
110 PRINT                                 <007>
121 SA=820:FOR I=SA TO SA+6:READ A:POKE I,
    A:NEXT I                              <073>
122 DATA 133,95,134,96,76,191,163         <179>
130 POKE 88,0:POKE 89,192:POKE 90,0:POKE 9
    1,192:POKE 780,0:POKE 781,160:SYS SA  <244>
140 POKE 88,0:POKE 89,0:POKE 90,0:POKE 91,
    0:POKE 780,0:POKE 781,224:SYS SA      <039>
150 POKE 1,53:POKE 42289,96:POKE 42290,228 <001>
160 FOR I=58464 TO 58554:READ A:POKE I,A:N
    EXT I                                 <100>
190 PRINT"{4DOWN,9SPACE}CHECKSUMMER AKTIVI
    ERT."                                 <247>
200 PRINT"{2DOWN}AUSSCHALTEN : POKE1,55"  <050>
210 PRINT"{DOWN}ANSCHALTEN{2SPACE}: POKE1,
    53":NEW                               <171>
320 DATA 160,2,169,0,133,2,177,95         <103>
330 DATA 240,15,201,32,208,3,200,208      <239>
340 DATA 245,24,101,2,133,2,76,110        <153>
350 DATA 228,192,4,48,241,198,214,165     <090>
360 DATA 214,72,162,3,169,32,157,1        <191>
370 DATA 4,189,183,228,32,210,255,202     <096>
380 DATA 16,242,166,2,169,0,32,205        <206>
390 DATA 189,169,62,32,210,255,104,133    <168>
400 DATA 214,32,108,229,169,141,32,210    <168>
410 DATA 255,76,128,164,92,72,32,201      <093>
420 DATA 255,170,104,144,1,138,96,9       <051>
430 DATA 60,18,19                         <195>
```

© 64'er

Der Checksummer für den C 64

CTRL steht für Control-Taste, so bedeutet [CTRL-A], daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

{DOWN}	Taste neben rechtem Shift, Cursor unten
{UP}	Shift-Taste & Taste neben rechtem Shift, Cursor hoch
{CLEAR}	Shift-Taste & 2. Taste ganz rechts oben
{INST}	Shift-Taste & Taste ganz rechts oben
{HOME}	2. Taste von ganz rechts oben
{DEL}	Taste ganz rechts oben
{RIGHT}	Taste ganz rechts unten
{LEFT}	Shift-Taste & Taste unten rechts
{SPACE}	Leertaste
{F1}	grauer Tastenblock rechts
{F3}	grauer Tastenblock rechts
{F5}	grauer Tastenblock rechts
{F7}	grauer Tastenblock rechts
{F2}	grauer Tastenblock rechts & Shift
{F4}	grauer Tastenblock rechts & Shift
{F6}	grauer Tastenblock rechts & Shift
{F8}	grauer Tastenblock rechts & Shift
{RETURN}	Shift-Taste & Return
{BLACK}	Control-Taste & 1
{WHITE}	Control-Taste & 2
{RED}	Control-Taste & 3
{CYAN}	Control-Taste & 4
{PURPLE}	Control-Taste & 5
{GREEN}	Control-Taste & 6
{BLUE}	Control-Taste & 7
{YELLOW}	Control-Taste & 9
{RVSON}	Control-Taste & 8
{RVOFF}	Control-Taste & 0
{ORANGE}	Commodore-Taste & 1
{BROWN}	Commodore-Taste & 2
{LIG.RED}	Commodore-Taste & 3
{GREY 1}	Commodore-Taste & 4
{GREY 2}	Commodore-Taste & 5
{LIG.GREEN}	Commodore-Taste & 6
{LIG.BLUE}	Commodore-Taste & 7
{GREY 3}	Commodore-Taste & 8

Wenn Sie sich erst einmal an die in Klartext geschriebenen Steuerzeichen gewöhnt haben, werden Sie den Vorteil dieser Schreibweise erkennen. Der zu dem jeweiligen Steuerzeichen gehörende Klartext ist so verfaßt, daß Sie leicht die Taste beziehungsweise die Tastenkombination finden, die Sie drücken müssen.

Die Steuerbefehle im Klartext

Hinweis: {13 SPACE} bedeutet 13mal die Leertaste drücken

MSE-Abtippen sicher und leicht gemacht

Ähnlich wie der »Checksummer« ist auch der MSE ein Hilfsmittel bei der Eingabe von Listings, diesmal jedoch bei reinen Maschinensprache-Programmen.

Im Gegensatz zum »Checksummer« aber ist die Eingabe nicht ohne den MSE möglich. Der MSE verringert die Tipparbeit um ein Drittel und schließt Fehleingaben vollkommen aus. Außerdem können Sie die DATAs blind eingeben, ohne andauernd auf den Bildschirm schauen zu müssen. Dies wird durch akustische Meldungen realisiert.

MSE ist ein Maschinenspracheditor, mit dem ein Vertippen ausgeschlossen ist. Eine abgetippte Zeile wird nur angenommen, wenn sie richtig ist. Eine Checksumme am Ende jeder Zeile prüft, ob die richtigen Werte in der richtigen Zeile an der richtigen Stelle stehen. Wenn nicht, ertönt ein Warnsignal, und man beseitigt den Fehler.

War die Zeile korrekt, erklingt ein Gong, und die nächste Zeilennummer wird ausgegeben. Damit ist also auch »blindes« Eintippen möglich; Sie können sich voll auf den Text konzentrieren.

So arbeitet man mit MSE

Laden und starten Sie MSE. Zuerst wird der Programmname und die Start- und Endadresse erfragt. **Diese Angaben entnehmen Sie dem Kopf des jeweiligen abgedruckten Listings.** MSE meldet sich dann mit der Zeilennummer der ersten Zeile. Wenn Sie die Zeile richtig eingegeben haben, erscheint die nächste Zeilennummer und so weiter bis zum Ende. Zum Schluß wird das fertige Programm mit »CTRL-S« auf Diskette oder Kassette abgespeichert. Dazu sind keine weiteren Angaben mehr erforderlich. Das Programm kann dann ganz normal wieder geladen und gestartet werden. Wenn Sie nicht alles auf einmal tippen wollen, können Sie jederzeit unterbrechen und den eingetippten Teil mit »CTRL-S« abspeichern. Wollen Sie weiterarbeiten, laden und starten Sie MSE wieder.

Geben Sie auf die Frage nach der Startadresse aber jetzt »L« ein, um Ihr Teilprogramm zu laden. Jetzt können Sie mit »CTRL-N« die Adresse eingeben, an der Sie weitertippen müssen. Wenn Sie sich nicht gemerkt haben, wie weit Sie gekommen sind, geben Sie nach dem Laden »CTRL-M« ein.

Auf die Frage nach der Startadresse antworten Sie mit der Anfangsadresse, die links in der Kopfzeile auf dem Bildschirm steht. Nun wird Ihr Programm aufgelistet. Mit »SPACE« wird das Listen fortgesetzt, mit »STOP« abgebrochen. Das Ende Ihres Programmteils erkennen Sie sehr einfach daran, daß nur noch der Wert »AA« in der Zeile steht. Die Adresse dieser Zeile müssen Sie anschließend mit »CTRL-N« eingeben. Das Programm ist nur mit »STOP/RESTORE« zu verlassen. Speichern Sie aber vorher unbedingt immer Ihren Text ab.

Hinweise zum Abtippen

Vor dem Abtippen oder späteren Wiederladen des MSE-Laders müssen Sie unbedingt folgende Zeile eingeben:

POKE 43,1:POKE 44,32:POKE 8192,0 NEW

Starten Sie das Programm mit RUN. Fehlerhafte Zeilen werden angezeigt und müssen korrigiert werden, bis der Lader zum »READY« durchläuft. Jetzt müssen Sie das fertige MSE-Programm abspeichern. Dazu brauchen Sie nur »RETURN« zu drücken, weil die erforderlichen Angaben schon auf dem Bildschirm stehen. (Kassettenbesitzer müssen in Zeile 343 die letzte Zahl in »1« abändern). Ab jetzt können Sie »MSE V1.0« direkt, also ohne den DATA-Lader, benutzen, MSE V1.0 wird ganz normal mit »,8« geladen (keine POKEs notwendig).

(N. Mann /D. Weineck/gk)

MSE-Befehle:

DEL	löscht die letzt Eingabe.
CTRL-S	speichert das eingetippte Programm ab.
CTRL-L	lädt ein Programm. Start- und Endadresse werden automatisch ermittelt.
CTRL-M	listet den Speicherinhalt. Abbruch mit STOP-Taste, weiter mit Leertaste.
CTRL-N	erlaubt die Eingabe einer neuen Adresse zum Weitertippen.
CTRL-P	gibt ein MSE-Listing auf dem Drucker aus.

```
1 REM *********************************    <208>
2 REM *       ++++ MSE - LADER    ++++  *   <183>
7 REM *********************************    <214>
8 :                                        <066>
9 :                                        <067>
10 DIM H(75) : FOR I=0 TO 9                <088>
20 H(48+I)=I : H(65+I)=I+10 : NEXT         <250>
30 FOR I=2048 TO 3755 : READ A$            <006>
40 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))  <063>
50 D=H(H)*16+H(L) : S=S+D : POKE I,D       <181>
60 A=A+1:IF A<9  THEN NEXT : A=-1          <111>
65 PRINT "ZEILE:";1000+Z;                  <012>
70 READ V : Z=Z+1 : IF V=S THEN 85         <210>
80 PRINT" PRUEFSUMMENFEHLER !";999+Z:STOP  <015>
85 IF A<0 THEN 341                         <067>
90 S=0 : A=0 : PRINT : NEXT : END          <053>
95 :                                       <153>
96 :                                       <154>
341 PRINT"{CLR}PO43,1:PO44,8:PO45,172:PO46
    ,14"                                   <170>
342 POKE 631,19:POKE 632,13:POKE 633,13:PO
    KE 198,3                               <233>
343 PRINT"{3DOWN}SAVE"CHR$(34)"MSE V1.0"CH
    R$(34)",8                              <041>
344 END                                    <217>
360 :                                      <163>
1000 DATA 00,0B,08,0A,00,9E,32,30,36, 339  <083>
1001 DATA 31,00,00,00,A2,08,A9,36,85, 575  <075>
1002 DATA A4,A9,08,85,A5,A9,00,85,A6, 1107 <189>
1003 DATA A9,B0,85,A7,A0,00,B1,A4,91, 1291 <190>
1004 DATA A6,C8,D0,F9,E6,A5,E6,A7,CA, 1817 <028>
1005 DATA D0,F2,A9,36,85,01,4C,00,B0, 1059 <180>
1006 DATA 20,D1,B1,A9,06,8D,21,D0,A9, 1144 <193>
1007 DATA 03,8D,20,D0,8D,86,02,A0,B3, 1000 <169>
1008 DATA A9,74,20,FF,B1,A0,B3,A9,B9, 1442 <238>
1009 DATA 20,FF,B1,A0,00,20,CF,FF,99, 1271 <233>
1010 DATA 01,02,C8,C9,0D,D0,F5,88,F0, 1246 <212>
1011 DATA D2,C0,0F,90,02,A0,0E,8C,00, 877  <150>
1012 DATA 02,20,EA,B1,A0,B3,A9,CF,20, 1192 <219>
1013 DATA FF,B1,20,8E,B4,85,FC,85,62, 1402 <237>
1014 DATA 20,8E,B4,85,FB,85,61,20,A7, 1167 <207>
1015 DATA B4,D0,20,A0,B3,A9,E5,20,FF, 1444 <234>
1016 DATA B1,20,8E,B4,85,60,20,8E,B4, 1114 <193>
1017 DATA 85,5F,20,A7,B4,D0,0A,A5,61, 1087 <213>
1018 DATA C5,5F,A5,62,E5,60,90,06,20, 1062 <183>
1019 DATA 43,B3,4C,3A,B0,A9,AA,A0,00, 1055 <222>
1020 DATA 91,FB,E6,FB,D0,02,E6,FC,20, 1601 <007>
1021 DATA 3F,B2,90,EF,4C,FB,B4,A2,02, 1295 <011>
1022 DATA 86,58,A9,A6,A0,9D,20,F2,B1, 1325 <226>
1023 DATA 20,E4,FF,F0,FB,C9,30,90,0C, 1411 <248>
```

Der MSE zum bequemen Abtippen von Assemblerprogrammen

```
1024 DATA C9,47,B0,08,C9,3A,90,0B,C9, 1071 <228>
1025 DATA 41,B0,07,C9,14,D0,0F,4C,0B, 779  <173>
1026 DATA B1,20,D2,FF,A6,58,95,F7,C6, 1522 <254>
1027 DATA 58,D0,D2,60,AE,8D,02,F0,26, 1197 <231>
1028 DATA C9,0C,D0,03,4C,0B,B6,C9,13, 913  <187>
1029 DATA D0,03,4C,8B,B5,C9,0D,D0,03, 1032 <228>
1030 DATA 4C,BA,B4,C9,10,D0,03,4C,68, 1050 <232>
1031 DATA B5,C9,0E,D0,06,20,5F,B4,4C, 993  <203>
1032 DATA 64,B1,4C,92,B0,A5,F9,20,02, 1123 <204>
1033 DATA B1,0A,0A,0A,0A,85,F9,A5,F8, 1012 <247>
1034 DATA 20,02,B1,05,F9,60,C9,3A,90, 964  <154>
1035 DATA 02,69,08,29,0F,60,A6,59,E0, 746  <153>
1036 DATA 08,90,1F,A6,58,E0,02,B0,06, 845  <155>
1037 DATA 20,D2,FF,4C,8E,B0,C6,59,A0, 1338 <017>
1038 DATA 14,A9,92,20,F2,B1,CA,D0,FA, 1446 <006>
1039 DATA 84,57,68,68,4C,8B,B1,A6,D3, 1196 <249>
1040 DATA E0,08,B0,03,4C,92,B0,20,D2, 1051 <200>
1041 DATA FF,A6,58,E0,02,90,09,C6,59, 1175 <242>
1042 DATA 20,D2,FF,C6,58,D0,F9,4C,8E, 1458 <040>
1043 DATA B0,48,4A,4A,4A,4A,20,59,B1, 842  <185>
1044 DATA 68,29,0F,C9,0A,90,02,69,06, 628  <163>
1045 DATA 69,30,4C,D2,FF,A2,FC,9A,20, 1294 <027>
1046 DATA D1,B1,20,48,B2,20,EA,B1,20, 1143 <217>
1047 DATA 9F,B2,A5,FC,20,4E,B1,A5,FB, 1457 <057>
1048 DATA 20,4E,B1,20,ED,B1,A9,3A,A0, 1120 <250>
1049 DATA 20,20,F2,B1,A9,00,85,59,20, 906  <145>
1050 DATA 8E,B0,20,ED,B1,A4,59,20,EF, 1288 <029>
1051 DATA B0,91,FB,C8,84,59,C0,08,90, 1337 <249>
1052 DATA EC,20,10,B2,A9,12,20,D2,FF, 1146 <251>
1053 DATA 20,8E,B0,20,EF,B0,C5,FF,F0, 1489 <048>
1054 DATA 0D,20,43,B3,A9,14,A0,14,20, 692  <155>
1055 DATA F2,B1,4C,A2,B1,A9,92,20,D2, 1391 <005>
1056 DATA FF,20,33,B2,20,E0,B2,20,3F, 1045 <203>
1057 DATA B2,90,9F,4C,8B,B5,A9,93,20, 1225 <010>
1058 DATA D2,FF,A2,00,A9,03,9D,00,D8, 1172 <011>
1059 DATA 9D,00,D9,9D,00,DA,9D,00,DB, 1125 <030>
1060 DATA E8,D0,EF,60,A9,0D,2C,A9,20, 1202 <029>
1061 DATA 4C,D2,FF,20,D2,FF,98,4C,D2, 1476 <069>
1062 DATA FF,20,E4,FF,F0,FB,60,84,5D, 1582 <069>
1063 DATA 85,5C,A0,00,B1,5C,F0,06,20, 932  <183>
1064 DATA D2,FF,C8,D0,F6,60,A5,FB,85, 1764 <075>
1065 DATA 5A,A0,00,84,5B,B1,FB,18,65, 1026 <254>
1066 DATA 5A,85,5A,90,02,E6,5B,06,5A, 876  <212>
1067 DATA 26,5B,C8,C0,08,90,EC,A5,5A, 1164 <028>
1068 DATA 65,5B,85,FF,60,18,A5,FB,69, 1221 <028>
1069 DATA 08,85,FB,90,02,E6,FC,60,A5, 1281 <020>
1070 DATA FB,C5,5F,A5,FC,E5,60,60,A0, 1541 <061>
1071 DATA B3,A9,FB,20,FF,B1,A0,01,B9, 1409 <053>
1072 DATA 00,02,20,D2,FF,CC,00,02,C8, 905  <202>
1073 DATA 90,F4,A9,10,ED,00,02,AA,20, 1014 <247>
1074 DATA ED,B1,CA,D0,FA,A5,62,20,4E, 1447 <073>
1075 DATA B1,A5,61,20,4E,B1,20,ED,B1, 1172 <013>
1076 DATA A5,60,20,4E,B1,A5,5F,20,4E, 918  <223>
1077 DATA B1,A9,9F,20,D2,FF,20,EA,B1, 1445 <065>
1078 DATA 24,5E,10,01,60,A9,12,20,D2, 672  <165>
1079 DATA FF,A2,28,20,ED,B1,CA,D0,FA, 1563 <095>
1080 DATA A9,92,4C,D2,FF,A5,D6,C9,16, 1458 <078>
1081 DATA B0,01,60,A9,A0,85,A4,A9,78, 1188 <013>
1082 DATA 85,A6,A9,04,85,A5,85,A7,A2, 1232 <018>
1083 DATA 13,A0,27,B1,A4,91,A6,88,10, 1022 <235>
1084 DATA F9,CA,F0,19,18,A5,A4,69,28, 1214 <039>
1085 DATA 85,A4,90,02,E6,A5,18,A5,A6, 1193 <018>
1086 DATA 69,28,85,A6,90,E0,E6,A7,4C, 1285 <038>
1087 DATA B6,B2,A9,91,4C,D2,FF,A9,0F, 1399 <097>
1088 DATA 8D,18,D4,A9,00,8D,05,D4,A9, 1073 <040>
1089 DATA F7,8D,06,D4,A9,11,8D,04,D4, 1149 <046>
1090 DATA A9,32,8D,01,D4,A9,00,8D,00, 883  <226>
1091 DATA D4,A0,80,20,09,B3,A9,10,8D, 1046 <009>
1092 DATA 04,D4,60,A2,FF,CA,D0,FD,88, 1528 <090>
1093 DATA D0,F8,60,A9,0F,8D,18,D4,A9, 1282 <068>
1094 DATA 2D,8D,05,D4,A9,A5,8D,06,D4, 1096 <066>
1095 DATA A9,21,8D,04,D4,A9,07,8D,01, 877  <243>
1096 DATA D4,A9,05,8D,00,D4,A0,FF,20, 1186 <053>
1097 DATA 09,B3,A9,20,8D,04,D4,A9,00, 915  <231>
1098 DATA 8D,01,D4,8D,00,D4,60,38,20, 891  <219>
1099 DATA F0,FF,8A,48,98,48,18,A0,06, 1119 <046>
1100 DATA A2,18,20,F0,FF,A0,B4,A9,0A, 1232 <057>
1101 DATA 20,FF,B1,20,12,B3,20,E4,FF, 1208 <045>
1102 DATA F0,FB,A2,1D,A9,14,20,D2,FF, 1368 <092>
1103 DATA CA,D0,FA,68,A8,68,AA,18,4C, 1306 <098>
1104 DATA F0,FF,0D,0D,0D,20,20,20,20, 662  <231>
1105 DATA 20,20,20,4D,41,53,43,48,49, 533  <170>
1106 DATA 4E,45,4E,53,50,52,41,43,48, 674  <205>
1107 DATA 45,20,2D,20,45,44,49,54,4F, 551  <197>
1108 DATA 52,20,0D,0D,20,20,20,20,20, 300  <149>
1109 DATA 20,20,20,56,4F,4E,20,4E,2E, 495  <224>
1110 DATA 4D,41,4E,4E,20,26,20,44,2E, 514  <221>
1111 DATA 57,45,49,4E,45,43,4B,00,0D, 531  <216>
1112 DATA 0D,0D,20,20,20,50,52,4F,47, 434  <197>
1113 DATA 52,41,4D,4D,4E,41,4D,45,20, 622  <227>
1114 DATA 3A,20,00,0D,0D,20,20,20,53, 295  <185>
1115 DATA 54,41,52,54,41,44,52,45,53, 682  <177>
1116 DATA 53,45,20,3A,20,24,00,0D,0D, 336  <194>
1117 DATA 20,20,20,45,4E,44,41,44,52, 526  <177>
1118 DATA 45,53,53,45,20,20,20,3A,20, 490  <172>
1119 DATA 24,00,92,05,20,50,52,4F,47, 531  <180>
1120 DATA 52,41,4D,4D,20,3A,20,00,12, 441  <195>
1121 DATA 20,20,2A,2A,2A,20,46,41,4C, 433  <211>
1122 DATA 53,43,48,45,20,45,49,4E,47, 614  <208>
1123 DATA 41,42,45,20,2A,2A,2A,20,20, 422  <193>
1124 DATA 92,00,0D,0D,2A,2A,2A,20,45, 399  <243>
1125 DATA 4E,44,45,20,2A,2A,2A,00,13, 392  <223>
1126 DATA 05,20,20,12,44,92,49,53,4B, 532  <189>
1127 DATA 20,4F,44,45,52,20,12,54,92, 610  <190>
1128 DATA 41,50,45,0D,00,13,20,20,49, 383  <181>
1129 DATA 2F,4F,20,2D,20,46,45,48,4C, 522  <247>
1130 DATA 45,52,00,20,D1,B1,20,48,B2, 851  <215>
1131 DATA A0,B3,A9,CF,20,FF,B1,20,8E, 1353 <115>
1132 DATA B4,85,FC,20,8E,B4,85,FB,C5, 1500 <115>
1133 DATA 61,A5,FC,E5,62,90,23,A5,FB, 1436 <098>
1134 DATA C5,5F,A5,FC,E5,60,B0,19,20, 1267 <097>
1135 DATA A7,B4,D0,14,60,20,A7,B4,F0, 1290 <065>
1136 DATA 0C,85,F9,20,A7,B4,F0,05,85, 1151 <066>
1137 DATA F8,4C,EF,B0,68,68,20,43,B3, 1225 <090>
1138 DATA 4C,5F,B4,20,CF,FF,C9,4C,D0, 1330 <146>
1139 DATA 09,20,D1,B1,20,48,B2,4C,0B, 796  <010>
1140 DATA B6,C9,0D,60,A9,00,85,5E,20, 920  <019>
1141 DATA 5F,B4,20,EA,B1,20,0D,B5,24, 980  <040>
1142 DATA 5E,30,05,20,E4,FF,F0,FB,20, 1185 <097>
1143 DATA E1,FF,F0,26,20,9F,B2,24,5E, 1257 <110>
1144 DATA 10,09,20,4E,B5,20,0D,B5,20, 574  <246>
1145 DATA 60,B5,20,33,B2,20,3F,B2,90, 955  <000>
1146 DATA D7,A0,B4,A9,28,20,FF,B1,20, 1260 <095>
1147 DATA E4,FF,C9,0D,D0,F9,A9,00,85, 1456 <141>
1148 DATA 5E,A5,61,85,FB,A5,62,85,FC, 1388 <131>
1149 DATA 20,E0,B2,4C,64,B1,A5,FC,20, 1236 <092>
1150 DATA 4E,B1,A5,FB,85,FF,20,4E,B1, 1346 <144>
1151 DATA A9,20,A0,3A,20,F2,B1,A0,00, 1030 <053>
1152 DATA 20,ED,B1,B1,FB,20,4E,B1,C8, 1361 <128>
1153 DATA C0,08,90,F3,20,ED,B1,24,5E, 1163 <090>
1154 DATA 30,03,A9,12,2C,A9,20,20,D2, 725  <255>
1155 DATA FF,20,10,B2,A5,FF,20,4E,B1, 1188 <123>
1156 DATA A9,92,20,D2,FF,4C,EA,B1,A9, 1468 <157>
1157 DATA FF,85,B8,85,B9,A9,04,85,BA, 1382 <143>
1158 DATA 20,C0,FF,A2,FF,4C,C9,FF,20, 1460 <165>
1159 DATA CC,FF,A9,FF,4C,C3,FF,20,5F, 1536 <215>
1160 DATA B4,A9,80,85,5E,20,4E,B5,20, 1027 <088>
1161 DATA 48,B2,A2,24,A9,2D,20,D2,FF, 1159 <121>
1162 DATA CA,D0,FA,20,EA,B1,20,EA,B1, 1546 <162>
1163 DATA 20,60,B5,4C,C1,B4,20,B8,B5, 1155 <093>
1164 DATA A6,5F,A4,60,A9,61,20,D8,FF, 1290 <131>
1165 DATA B0,0A,20,B7,FF,29,BF,D0,03, 1099 <135>
1166 DATA 4C,FB,B4,A9,01,20,C3,FF,20, 1191 <131>
1167 DATA 68,B6,A0,B4,A9,4F,20,FF,B1, 1338 <147>
1168 DATA 20,F9,B1,4C,FB,B4,20,68,B6, 1283 <130>
1169 DATA A9,37,A0,B4,20,FF,B1,20,F9, 1309 <126>
1170 DATA B1,A2,08,C9,44,F0,06,A2,01, 1025 <079>
1171 DATA C9,54,D0,F1,A9,01,A8,20,BA, 1290 <123>
1172 DATA FF,A0,00,E0,01,F0,1A,A9,40, 1139 <110>
1173 DATA 8D,20,02,A9,3A,8D,21,02,B9, 763  <050>
1174 DATA 01,02,99,22,02,C8,CC,00,02, 598  <013>
1175 DATA 90,F4,C8,C8,D0,0C,B9,01,02, 1196 <121>
1176 DATA 99,20,02,C8,CC,00,02,D0,F4, 1045 <092>
1177 DATA 98,A2,20,A0,02,4C,BD,FF,20, 1060 <119>
1178 DATA B8,B5,A5,BA,C9,08,90,33,A6, 1286 <146>
1179 DATA B9,86,57,A9,01,20,C3,FF,A9, 1227 <136>
1180 DATA 60,85,B9,20,C0,FF,B0,28,A5, 1274 <126>
1181 DATA BA,20,B4,FF,A5,B9,20,96,FF, 1440 <174>
1182 DATA 20,A5,FF,85,61,A5,90,4A,4A, 1139 <128>
1183 DATA B0,13,20,A5,FF,85,62,20,AB, 1081 <112>
1184 DATA FF,A5,57,85,B9,A9,00,20,D5, 1239 <141>
1185 DATA FF,90,03,4C,A3,B5,86,5F,B4, 1183 <144>
1186 DATA 60,A5,BA,C9,01,D0,0A,AD,3D, 1101 <149>
1187 DATA 03,85,61,AD,3E,03,85,62,4C, 778  <063>
1188 DATA FB,B4,A9,13,20,D2,FF,A2,1C, 1306 <168>
1189 DATA 20,ED,B1,CA,D0,FA,60, 1202       <104>
```

Listing von MSE (Schluß)

Sternen-himmel

Um über die Positionen von Fixsternen und Planeten informiert zu sein, schrieb Horst Hinkelmann dieses Programm für den C 64 mit Simons Basic.

Nach dem Start des Programms werden zunächst das aktuelle Datum sowie die aktuelle Uhrzeit (MEZ) und die geografischen Koordinaten des Beobachtungsortes eingegeben. Dabei müssen nördliche Breiten und westliche Längen positiv, südliche Breiten und östliche Längen negativ angegeben werden. Nun kann man den Beobachtungszeitpunkt wählen, wenn sich dieser vom bereits im Programm festgelegten Zeitpunkt unterscheidet.

Es folgt eine kurze Erläuterung des Programmablaufs und das Zeichnen der Sternkarte. Mit den Cursor-Tasten kann über ein Kreuz jedes Objekt der Sternkarte angesteuert werden. Die Bewegungsgeschwindigkeit ist abhängig von der Dauer des Tastendrucks; das heißt je länger eine Cursortaste gedrückt wird, desto schneller bewegt sich das Kreuz. Drückt man nach der Wahl die Funktionstaste F1, gibt der Computer den Namen des Sterns oder Planeten, der dem Kreuz am nächsten ist, an. Dabei wird der Cursor genau auf das Objekt positioniert. Möchte man wissen, um welches Sternbild es sich bei einer bestimmten Konstellation handelt, bringt man das Kreuz auf das betreffende Sternbild und drückt die Funktionstaste F3. Der Computer positioniert den Cursor auf den nächsten Stern des Sternbildes und zeigt den Namen des Sternes und des Sternbildes an. Das gesamte Sternbild blinkt. Es besteht auch die Möglichkeit, einen Stern, ein Sternbild oder einen Planeten nach dem Namen suchen zu lassen. Dazu gibt man einfach mit der Tastatur den Namen oder einen Teil des Namens des gesuchten Objekts ein und schließt die Eingabe mit der RETURN-Taste ab. Der Computer positioniert das Kreuz auf dieses Objekt und gibt den vollständigen Namen aus. Ist das gesuchte Objekt nicht gespeichert oder zum Beobachtungszeitpunkt am Beobachtungsort nicht zu sehen, erhält man eine entsprechende Fehlermeldung.

Will man die Sternkarte für einen anderen Beobachtungsort oder einen anderen Beobachtungszeitpunkt erstellen lassen, drückt man die Funktionstaste F7.

Mit F5 wird die Sternkarte auf einen MPS 801 gedruckt.

Programmerläuterung

Auf die Formeln, die den Berechnungen zugrunde liegen, soll verzichtet werden. Diese Berechnungen sind (zumindest für Mond- und Planetenpositionen) ziemlich kompliziert und können bei Interesse jedem größeren Werk über Astronomie entnommen werden.

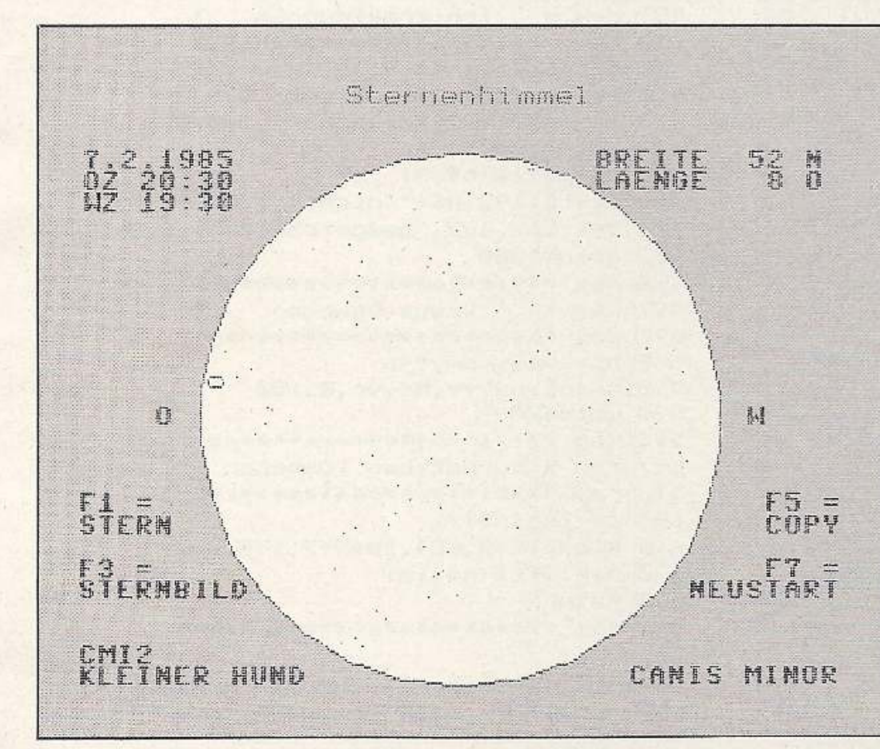
Von programmtechnischem Interesse dürften allerdings die Suchroutinen sein. Deren Funktionsweise soll deshalb erläutert werden.

Zum Suchen werden die folgenden Arrays verwendet:

P$	Namen der Planeten
Z$	Namen der Sternbilder
P	Bildschirmkoordinaten der Planeten
Z%	Bildschirmkoordinaten der Fixsterne
Z	Zeiger auf Fixsterne (für jedes Sternbild)

Suchen Fixstern/Planet (Funktionstaste F1)

Zunächst werden die Bildschirmkoordinaten des Kreuzes (X,Y) mit denen der Planeten (X mit P(0), P(2), ..., Y mit P(1), P(3), ...) verglichen. Bei Übereinstimmung ist XX=0 und die Suche kann abgebrochen werden. In Zeile 42090 verzweigt das Programm nach 45000, wo der Name des Planeten geschrieben wird. Liegt keine Übereinstimmung vor, wird der Planet mit dem geringsten Abstand gespeichert (XP = Abstand^2, ZP = Kennziffer des Planeten). Nun werden die Fixsterne in gleicher Weise überprüft. Der Stern mit dem geringsten Abstand wird gespeichert (Z = Kennziffer des Sterns; Zeile 44020). Wird kein Stern mit geringerem Abstand als der nächste Planet gefunden, wird der gespeicherte Planet genannt (Zeilen 44070 bis 45010). Dabei wird ab der Zeile 500 das Kreuz exakt auf den Planeten beziehungsweise Stern positioniert und dessen Bezeichnungen ausgegeben. Der Name eines Planeten ergibt sich aus dem Array P$ mit dem Index ZP. Der Name eines Sternes wird durch berechne-



Hardcopy des Sternenhimmels von Hamm (zirka 52° nördliche Breite und 8° östliche Länge), wobei der Beobachtungszeitpunkt der 7. Februar 1985, 19.30 GMT (Weltzeit) beziehungsweise 20.30 MEZ (Ortszeit) ist. Wie bei Sternkarten üblich, ist Osten links, Westen rechts, Norden unten und Süden oben. Der Mond steht tief im Osten.

tes RESET (Zeile 44080) aus den DATA-Zeilen 60000 bis 61240 ermittelt. Die Schreibroutine befindet sich ab Zeile 300.

Suchen Sternbild (Funktionstaste F3)

Zunächst wird die Suchroutine für Sterne mit Ausnahme des Abschnitts für die Planeten durchlaufen. Der gefundene nächste Stern bestimmt das Sternbild. Die Kennziffer Z des Sterns wird in Z1, die letzte Ziffer des gelesenen Datensatzes für den Stern (D; Zeile 44090) wird in Z gespeichert. Das Programm verzweigt nun in die Blink-Routine (Zeile 48060 bis 48250). Dort werden die zum Sternbild gehörenden Sterne (Punkte) abwechselnd gesetzt und gelöscht (Zeile 48170).

Suchen nach Namen

Wird ein Name, oder ein Teil eines Namens eingegeben, verzweigt das Programm in 40100 nach Zeile 47000. Das komplizierte Löschen des zu überschreibenden Textes (GOSUB 340, GOSUB 360 in Zeile 47030) ist erforderlich, da Simons Basic Text mit den auf dem Grafik-Bildschirm gesetzten Punkten ODER- verknüpft. Aus diesem Grund wird der vorige Text erst gelöscht. Nun wird geprüft, ob ein Planetenname mit der eingegebenen Bezeichnung übereinstimmt (47060 bis 47090). Ist dies der Fall, werden die Bildschirmkoordinaten des Planeten zu denen des Kreuzes (Zeile 47120). Ist der Planet zur Beobachtungszeit nicht sichtbar, dann ist X = 0 und eine Fehlermeldung wird ausgegeben (47140). Ansonsten wird das Kreuz auf den Planeten positioniert und in die Eingaberoutine gesprungen (47150). Handelt es sich bei der eingegebenen Bezeichnung um keinen Planeten, wird die Suchroutine in 48000 fortgesetzt. Dort werden zunächst die Namen der Sternbilder (deutsch und lateinisch) auf Übereinstimmung untersucht. Wird keine Übereinstimmung gefunden, springt das Programm in die Zeile 49000. Hier werden nun die Sternbezeichnungen eingelesen und mit der Eingabe verglichen. Kommt es zu keiner Übereinstimmung, wird die Fehlermeldung »... nicht gespeichert« ausgegeben und das Kreuz in die linke obere Ecke des Bildschirms gesetzt (Zeile 49050). Darauf kehrt das Programm in die Eingaberoutine (ab Zeile 40000) zurück.

(Horst Hinkelmann/hm)

Programmstruktur von Sternenhimmel

Zeilen	Bedeutung
100 - 120	Berechnung des Stundenwinkels
130 - 140	Berechnung Bogenmaß
150 - 190	Koordinaten-Transformation
200 - 240	Berechnung Gradmaß
296 - 395	Text auf Grafikbildschirm überschreiben
396 - 480	Ausgabe Fehlermeldungen
496 - 520	Kreuz (Cursor) bewegen
596 - 630	Buchstaben auf Grafikbildschirm löschen
996 - 2190	Bildschirmmaske erstellen
4996 - 5260	Definition Kreuz als Sprite
9996 - 15080	Stellung Erde/Sonne/Mond berechnen
19996 - 20160	Stellung Planeten berechnen
24996 - 25080	Stellung Fixsterne berechnen
29996 - 30050	Rektaszension und Deklination berechnen
39996 - 49100	Eingabeschleife
49996 - 56050	Programmbeginn und Funktionsdefinitionen:
52010 - 52020	arc sin
52030 - 52040	arc cos
52050 - 52060	Modulo 360
57000 - 57110	Hardcopy
58000 - 58020	Neustart des Programms
60000 - 61240	Daten Fixsterne (Rektaszension, Deklination, Name, Nummer des Sternbildes)
62000 - 62230	Daten Sternbilder (Name lateinisch, Name deutsch, Zahl Sterne —1, Beginn der Sterndaten DATA-Zeile 60000 + X·10)
62500 - 62700	Daten Planeten (Name, Bahnelemente)

```
0 rem *******************************
1 rem *                             *
2 rem *        sternenhimmel        *
3 rem *                             *
4 rem *      horst  hinkelmann      *
5 rem *        nicolaistr. 6        *
6 rem *         4700 hamm 1         *
7 rem *     telefon  02385/1653     *
8 rem *                             *
9 rem *******************************
10 goto50000
96 rem **********************
97 rem *    stern setzen    *
98 rem **********************
100 rem ** stundenwinkel **
120 sw=fnmo(ar-re-1)
130 rem ** bogenmass **
140 sw=sw*p1
150 rem ** koordinaten-transformation **
160 h=fnas(sb*sin(de)+cb*cos(de)*cos(sw)
)
170 ifh<0thenx=0:y=0:return
180 a=fnac((sin(de)-sb*sin(h))/(cb*cos(h
)))
190 ifsw<pithena=-a
200 rem ** gradmass **
210 r=99-h*198/pi:w=-a-pi/2
220 x=int(cos(w)*r*1.1+159.5)
230 y=int(sin(w)*r+99.5)
240 plotx,y,1
296 rem **********************
297 rem * text ueberschreiben *
298 rem **********************
300 n$=left$(n$,13)
310 ifn3$<>n$thengosub340
320 ifn4$<>n1$thengosub360
330 return
340 n3$=n$:block0,184,103,191,0
350 text0,184,n$,1,0,8:return
360 n4$=n1$:block0,192,119,199,0
370 block216,192,319,199,0
380 text0,192,n1$,1,0,8
390 text319-8*len(n2$),192,n2$,1,0,8
395 return
396 rem **********************
397 rem *   fehlermeldungen   *
398 rem **********************
400 n1$=left$(n1$,9)
410 text0,192,n1$+" nicht",1,0,8
420 text232,192,"sichtbar",1,0,8
430 goto41000
450 n$=left$(n$,9)
460 text0,192,n$+" nicht",1,0,8
470 text232,192,"gespeichert",1,0,8
480 goto41000
496 rem **********************
497 rem *    kreuz bewegen    *
498 rem **********************
500 hn=x+19:vn=y+46
510 mmob1,hv,vv,hn,vn,0,100
520 goto300
596 rem **********************
597 rem * buchstaben loeschen *
598 rem **********************
600 le=len(n$)-1
610 blockle*8,184,le*8+7,199,0
620 n$=left$(n$,le)
630 return
996 rem **********************
```

Listing zu Sternenhimmel. Geben Sie das Listing nur dann ein, wenn Sie vorher Simons Basic geladen haben.

```
997 rem * bildschirm erstellen *
998 rem ***********************
1000 hires7,6
1010 circle160,100,110,100,1
1020 hu=0:ifb<0thenhu=200
1030 ifb<0thencircle160,hu+b/9*10,2,2,1
1040 text1,0,mid$(str$(t),2)+"."+mid$(st
r$(m),2)+"."+mid$(str$(j),2),1,0,8
1050 text1,8,"oz"+oz$,1,0,8
1060 text1,16,"wz"+wz$,1,0,8
1070 t$=mid$(str$(int(b+.5)),2,4)
1080 sp$=right$("   ",3-len(t$))
1090 text216,0,"breite "+sp$+t$,1,0,8
1100 t$=mid$(str$(int(l+.5)),2,4)
1110 sp$=right$("   ",3-len(t$))
1120 b$="n":ifb<0thenb$="s"
1130 text304,0,b$,1,0,8
1140 text216,8,"laenge "+sp$+t$,1,0,8
1150 l$="w":ifl<0thenl$="o"
1160 text304,8,l$,1,0,8
1170 text32,96,"o",1,0,8
1180 text280,96,"w",1,0,8
1190 gosub2010
1200 gosub2110
1210 text288,128,"f5 =",1,0,8
1220 text288,136,"copy",1,0,8
1230 text288,152,"f7 =",1,0,8
1240 text256,160,"neustart",1,0,8
1250 return
2000 block0,128,39,143,0
2010 text0,128,"f1 =",1,0,8
2020 text0,136,"stern",1,0,8
2030 return
2050 block0,128,39,143,0
2060 text0,128,"sucht",1,0,8
2070 text0,136,"stern",1,0,8
2080 return
2100 block0,152,39,159,0
2110 block0,160,71,167,0
2120 text0,152,"f3 =",1,0,8
2130 text0,160,"sternbild",1,0,8
2140 return
2150 block0,152,39,159,0
2160 block0,160,71,167,0
2170 text0,152,"sucht",1,0,8
2180 text0,160,"sternbild",1,0,8
2190 return
4996 rem **********************
4997 rem *    hires kreuz      *
4998 rem **********************
5000 design 0,32*64+49152
5020 @.....b.................
5030 @.....b.................
5040 @.....b.................
5050 @.....b.................
5060 @b.b.b.b.b.b............
5070 @.....b.................
5080 @.....b.................
5090 @.....b.................
5100 @.....b.................
5110 @.......................
5120 @.......................
5130 @.......................
5140 @.......................
5150 @.......................
5160 @.......................
5170 @.......................
5180 @.......................
5190 @.......................
5200 @.......................
5210 @.......................
5220 @.......................
5230 mob set 1,32,0,1,0
5240 hn=179:vn=145
5250 mmob1,hv,vv,hn,vn,0,0
5260 return
9996 rem *********************
9997 rem *  erde/sonne/mond  *
9998 rem *********************
10000 rem ** erde **
10100 be=fnmo(tg*.985609121+99.18)
10200 el=fnmo(be+sin((be-102.2)*p1)*1.84
5)
10300 ea=1+sin((el-192.2)*p1)*.0167
11000 rem ** sonne **
11200 ls=fnmo(el+180)
12000 rem ** mond **
12010 lm=tg*13.1763976+51.23
12020 pm=tg*.111399014+208.9
12030 km=372.1-tg*.052953643
12040 lm=fnmo(lm)
12050 pm=fnmo(pm)
12060 km=fnmo(km)
12070 am=lm-pm
12080 km=km-sin(as*p1)*.16
12090 ms=(lm-ls)*2-am
12100 am=am+sin(ms*p1)*1.27388889-sin(as
*p1)*(.18638889+.36)
12110 lm=lm+sin(ms*p1)*1.27388889-sin(as
*p1)*.18638889+sin(am*p1)*6.28833333
12120 m1=lm-ls
12130 lm=lm+sin(m1*2*p1)*.658333333
12140 m2=lm-km
12150 lm=lm-sin(m2*2*p1)*.12
12160 bm=sin(m2*p1)*5.14539
12170 m3=(lm-ls)*2-m2
12180 bm=bm+sin(m3*p1)*.15
15000 n$="sonne":al=ls:ab=0:gosub30000
15010 ifx=0goto15040
15020 circlex,y,4,3,1
15030 paintx+1,y,1
15040 n$="mond":al=lm:ab=bm:gosub30000
15050 ifx=0goto15080
15060 plotx,y,0
15070 circlex,y,4,3,1
15080 return
19996 rem *********************
19997 rem *      planeten     *
19998 rem *********************
20000 reset62500
20010 forz=0to5
20020 readn$,tb,ep,ph,mp,e,kn,i,ae
20030 p$(z)=n$
20040 ml=fnmo(tb*tg+ep)
20050 wl=ml+sin((ml-ph)*p1)*mp
20060 sp=ae+sin((wl-ph-90)*p1)*e*ae
20070 ws=fnmo(c+el-wl)*p1:si=sin(ws)
20080 fl=ea/sp-cos(ws)
20090 we=atn(si/fl)*p2
20100 al=fnmo(el+we-180*(fl>=0))
20110 wt=sin((wl-kn)*p1)*i
20120 ab=atn(tan(wt*p1)*abs(sin(we*p1)/s
i))*p2
20130 gosub30000
20140 p(2*z)=x:p(2*z+1)=y
20150 next
20160 return
24996 rem *********************
24997 rem *     fixsterne     *
24998 rem *********************
25000 reset60000
25010 fors=1toi1
25020 readre,de,n$,a
25030 de=de*p1
```

```
25040 n1$=z$(a,1):n2$=z$(a,0)
25050 gosub100
25060 z%(s)=x:z%(s+i2)=y
25070 next
25080 return
29996 rem ********************
29997 rem *  rektas./deklin.  *
29998 rem ********************
30000 sn=sin(ab*p1):cs=cos(ab*p1)
30010 sl=sin(al*p1):cl=cos(al*p1)
30020 de=fnas(ec*sn+es*cs*sl)
30030 re=2*atn((ec*cs*sl-es*sn)/(cos(de)
+cs*cl))
30040 re=fnmo(re*p2)
30050 goto100
39996 rem ********************
39997 rem *     eingaben      *
39998 rem ********************
40000 getg$:ifg$=""thenv=1:goto40000
40010 p=asc(g$)
40020 ifp=17thenvn=vn+v:goto41000
40030 ifp=145thenvn=vn-v:goto41000
40040 ifp=29thenhn=hn+v:goto41000
40050 ifp=157thenhn=hn-v:goto41000
40060 ifp=133thengosub2050:goto42000
40070 ifp=134thengosub2150:goto42000
40080 ifp=135goto57000
40090 ifp=136goto58000
40100 ifp>64andp<91goto47000
40110 goto40000
41000 v=v+.5
41010 mmob1,hn,vn,hn,vn,0,0
41020 goto40000
42000 x=int(hn-19):y=int(vn-46)
42010 xm=200000
42020 ifp=134goto44000
42030 fora=0to10step2
42035 f=p(a)-x:h=p(a+1)-y
42040 xx=f*f+h*h
42050 ifxx<xmthenxm=xx:zp=a
42060 ifxx=0thena=12
42070 next
42080 n$=p$(zp/2):n1$=" ":n2$=" "
42090 ifxx=0goto45000
44000 xp=xm
44010 fora=0toi1
44015 f=z%(a)-x:h=z%(a+i2)-y
44020 xx=f*f+h*h
44030 ifxx<xmthenxm=xx:z=a
44040 ifxx=0thena=i2
44050 next
44060 x=p(zp):y=p(zp+1)
44070 ifxp=xmgoto45000
44080 x=z%(z):y=z%(z+i2)
44090 reset60000+z*10
44100 readre,de,n$,d
44110 n1$=z$(d,1):n2$=z$(d,0)
44120 ifp=134thenz1=z:z=d:gosub2100:goto
48060
45000 gosub2000
45010 hv=hn:vv=vn:gosub500:goto40000
47000 n$=g$:n1$=" ":n2$=" "
47010 gosub340:gosub360
47020 getg$:ifg$=""goto47020
47030 ifasc(g$)=20thengosub600:goto47020
47040 ifasc(g$)=13goto47060
47050 n$=n$+g$:gosub350:goto47020
47060 p=0:z=-1
47070 fors=0to5
47080 ifplace(n$,p$(s))thenz=s:s=5
47090 next
47100 ifz<0goto48000
47110 hv=hn:vv=vn
47120 x=p(2*z):y=p(2*z+1)
47130 n1$=p$(z):n2$=" "
47140 ifx=0goto400
47150 gosub500:goto40000
48000 fors=0toi3
48010 ifplace(n$,z$(s,0))thenz=s:s=i3
48020 ifplace(n$,z$(s,1))thenz=s:s=i3
48030 next
48040 ifz<0goto49000
48050 z1=-1
48060 a1=z(z,1):a2=a1+z(z,0)
48070 ifz1>=0goto48120
48080 fora=a1toa2
48090 ifz%(a)thenz1=a:a=a2
48100 next
48110 ifz1<0thenn1$=z$(z,1):goto400
48120 f=0:hv=hn:vv=vn
48130 n1$=z$(z,1):n2$=z$(z,0)
48140 x=z%(z1):y=z%(z1+i2):gosub500
48150 fora=a1toa2
48160 x=z%(a)
48170 ifxthenplotx,z%(a+i2),f
48180 next
48190 f=1-f
48200 iff=1goto48150
48210 getg$:p=asc(g$+chr$(0))
48220 ifpgoto40020
48240 fora=1to1000*f+10:next
48250 goto48150
49000 reset60000
49010 fors=1toi1
49020 readre,de,s$,a
49030 ifplace(n$,s$)thenz=s:s=i2
49040 next
49050 ifz<0thenhn=x+19:vn=y+46:goto450
49060 ifz%(z)=0thenn1$=s$:goto400
49070 hv=hn:vv=vn
49080 x=z%(z):y=z%(z+i2)
49090 n$=s$:n1$=z$(a,1):n2$=z$(a,0)
49100 gosub500:goto40000
49996 rem ********************
49997 rem *  programmbeginn  *
49998 rem ********************
50000 cset1
50010 ifpeek(900)>0andpeek(900)<11then50
110
50020 print"Heutiges Datum"
50030 input"Jahr ";j$
50035 j$="19"+right$(j$,2)
50040 input"Monat";m$
50050 input"Tag  ";t$
50060 a$=t$+"."+m$+"."+j$
50070 fora=1tolen(a$)
50080 poke900+a,asc(mid$(a$,a,1))
50090 poke900,a
50100 next
50110 t$=left$(ti$,4)
50120 print"Uhrzeit HHMM"
50130 printtab(8)t$
50140 print""tab(6);:inputt1$:
50150 ift$<>t1$thenti$=right$("0"+t1$+"0
0",6)
50160 a$=""
50170 fora=0topeek(900)
50180 a$=a$+chr$(peek(900+a))
50190 next
50200 i1=124:i2=125:i3=22:f=1
50210 pi:p1=pi/180:p2=180/pi:c=360
50220 dimz$(i3,1),z(i3,1),p(11),z%(2*i2)
50300 print""
50301 printtab(11)"´###############%"
```

Listing Sternenhimmel (Fortsetzung)


64ER ONLINE

```
50302 printtab(11)"' Sternenhimmel %"
50304 printtab(11)"'$$$$$$$$$$$$$$$%"
50310 print"Breite  90 (noerdl) bis  -90
 (suedl.)"
50320 print"Laenge 180 (westl.) bis -180
 (oestl.)"
50330 print"Vorgegeben sind die Werte vo
n Hamm und  die heutige Zeit."
50340 print"Breite      "52:print""tab(10
);:inputb
50350 ifabs(b)>=90then50340
50360 print"Laenge       ";-8:print""tab(
10);:inputl
50370 ifabs(l)>180then50360
50380 sb=sin(b*p1):cb=cos(b*p1)
50390 print"Jahr         "right$(a$,4)
50400 print""tab(10);:inputj
50410 t=int(val(mid$(a$,2)))
50420 m=int(val(mid$(a$,len(str$(t))+2,2
)))
50430 print"Monat        "m
50440 print""tab(10);:inputm
50450 ifm>12orm<1then50440
50460 print"Tag          ";t
50470 print""tab(10);:inputt
50480 ift<1ort>31then50470
50490 wz=int(val(ti$)/100)-100
50495 ifwz<100thenwz=wz+2400
50500 print"Weltzeit     hhmm"
50505 printtab(12)right$(str$(wz),4)
50510 print""tab(10);:inputwz$:wz=val(wz
$)/100
50520 ifwz<0orwz>24goto50510
50530 zt=int(wz)+frac(wz)/.6
50600 i=m<3
50610 k=t+int((153*m-11*i-162)/5)+int((1
461*j+i)/4)+(j>=0)*366
50620 ifk>577736thenk=k-int((int((j+i)/1
00)*3-5)/4)
50630 ta=k-693596:tg=k-711858+zt/24:i=ta
/36525
50640 e=23.452294-i*.013125-i*i*1.639e-6
+i*i*i*5.028e-7
50650 es=sin(e*p1):ec=cos(e*p1)
52000 rem ** funktionen definieren **
52010 rem **  arcsin  **
52020 deffnas(x)=atn(x/sqr(1-x*x))
52030 rem **  arccos  **
52040 deffnac(x)=pi/2-atn(x/sqr(1-x*x))
52050 rem **  modulo  **
52060 deffnmo(x)=x-int(x/c)*c
54000 rem ** zeit **
54010 wz$=mid$(str$(int(frac(wz)*100+.5)
),2)
54020 wz$=right$("0"+wz$,2)
54030 wz$=right$(" "+str$(int(wz)),3)+":
"+wz$
54040 lo=int(l/15)*15
54050 oz=fnmo(zt*15-lo)/15
54060 oz$=right$(" "+str$(int(oz)),3)+ri
ght$(wz$,3)
54200 rem aries
54210 ar=zt*360.985647/24+frac(ta/1461)*
1440.02509
54220 ar=ar+int(ta/1461)*.0307572+99.201
8973
54230 ar=fnmo(ar)
54300 rem daten sternbilder
54310 reset62000
54320 fora=0toi3
54330 readz$(a,0),z$(a,1),z(a,0),z(a,1)
54340 next
```

```
54350 ifwhgoto56000
55000 rem ** anleitung **
55010 print"Dieses Programm zeichnet ein
e Sternen-"
55020 print"karte mit der Sonne, dem Mon
d, den"
55030 print"Planeten und den Fixsternen.
":print
55040 print"Die Namen der Sterne oder de
ren"
55050 print"Standorte koennen wie folgt
gesucht"
55060 print"werden:":print
55080 print"Mit den Cursor-Tasten das Kr
euz auf"
55090 print"oder in die Naehe des Sterne
s fuehren"
55100 print"und f1 druecken.":print
55110 print"Wenn das ganze Sternzeichen
gesucht"
55120 print"wird, dann f3 druecken.":pri
nt
55130 print"Wenn der Name oder ein Teil
des Namens"
55140 print"eines Sterns oder Sternbilde
s eingegeben";
55150 print"wird, geht das Kreuz alleine
 auf den":print"Stern"
55160 print"Bei einem Sternbild blinken
die dazu-"
55170 print"gehoerigen Sterne.":print
55180 print"Weiter = Taste druecken!";
55200 poke198,0:wait198,1
56000 gosub 1000:rem bildschirm
56010 gosub10000:rem sonne mond
56020 gosub20000:rem planeten
56030 gosub25000:rem fixsterne
56040 gosub 5000:rem kreuz
56050 goto40000
57000 rem ** hardcopy **
57100 copy
57110 goto40000
58000 rem ** neustart **
58010 nrm:cset1:mob off 1:wh=1
58020 goto50300
60000 rem ** daten fixsterne **
60010 data037.8,89.3,polaris,0
60020 data269.8,86.0,umi2,0
60030 data252.5,82.2,umi3,0
60040 data246.3,75.5,umi4,0
60050 data238.5,78.0,umi5,0
60060 data230.2,71.8,pherkad,0
60070 data222.7,74.2,kochab,0
60080 data206.9,49.3,benetnasch,1
60090 data201.0,54.9,mizar,1
60100 data193.5,56.0,alioth,1
60110 data183.9,57.0,megrez,1
60120 data178.5,53.7,phekda,1
60130 data165.9,61.8,dubhe,1
60140 data165.5,56.4,merak,1
60150 data28.6,63.7,cas1,2
60160 data21.5,60.2,cas2,2
60170 data14.2,60.7,cas3,2
60180 data10.1,56.5,schedir,2
60190 data2.3,59.2,caph,2
60200 data  3.3,15.2,algenib,3
60210 data346.2,15.2,markab,3
60220 data345.9,28.1,scheat,3
60230 data311.6,34.0,cyg1,4
60240 data310.4,45.3,deneb,4
60250 data305.6,40.3,schedir,4
60260 data296.2,45.1,cyg4,4
```

```
60270 data292.7,28.0,albireo,4
60280 data302.8,-0.8,aql1,5
60290 data297.7,08.9,atair,5
60300 data296.6,10.6,aql3,5
60310 data292.0,03.5,aql4,5
60320 data286.6,-4.9,aql5,5
60330 data286.4,13.9,aql6,5
60340 data284.7,32.7,lyr1,6
60350 data283.0,37.0,lyr2,6
60360 data282.5,33.4,lyr3,6
60370 data280.2,38.8,lyr4,6
60380 data279.2,38.8,wega,6
60390 data265.6,-39.0,sco1,7
60400 data264.3,-43.0,sco2,7
60410 data263.4,-37.1,sco3,7
60420 data262.7,-37.3,sco4,7
60430 data252.5,-34.3,sco5,7
60440 data249.0,-28.2,sco6,7
60450 data247.3,-26.4,antares,7
60460 data241.4,-19.8,acrab,7
60470 data240.1,-22.6,sco9,7
60480 data239.7,-26.1,sco10,7
60490 data228.9,33.3,boo1,8
60500 data225.0,41.0,boo2,8
60510 data221.2,27.1,boo3,8
60520 data218.0,38.3,boo4,8
60530 data218.0,30.5,boo5,8
60540 data213.9,19.2,arktur,8
60550 data201.3,-11.2,spika,9
60560 data198.0, -7.5,vir2,9
60570 data195.5, 11.0,vir3,9
60580 data193.9,  3.4,vir4,9
60590 data190.0, -1.0,vir5,9
60600 data185.0,  0.0,vir6,9
60610 data177.0,  2.0,vir7,9
60620 data191.9,-59.7,cru1,10
60630 data187.8,-57.1,cru2,10
60640 data186.7,-63.1,cru3,10
60650 data183.8,-58.8,cru4,10
60660 data177.3,14.5,denebola,11
60670 data168.6,15.4,leo2,11
60680 data168.5,20.5,leo3,11
60690 data155.0,19.9,leo4,11
60700 data152.1,12.0,regulus,11
60710 data151.8,16.8,leo7,11
60720 data146.5,23.8,leo6,11
60730 data116.3,28.0,pollux,12
60740 data113.7,31.9,kastor,12
60750 data101.3,12.9,gem3,12
60760 data101.0,25.1,gem4,12
60770 data 99.4,16.4,gem5,12
60780 data 95.7,22.5,gem6,12
60790 data111.0,-29.3,cma1,13
60800 data107.1,-26.4,cma2,13
60810 data104.7,-29.0,cma3,13
60820 data101.3,-16.7,sirius,13
60830 data 98.2,-18.0,cma5,13
60840 data90.0,37.2,aur1,14
60850 data89.9,45.0,aur2,14
60860 data79.2,46.0,capella,14
60870 data75.5,43.8,aur4,14
60880 data74.3,33.2,aur5,14
60890 data88.8, 7.4,beteigeuze,15
60900 data86.9,-9.1,ori2,15
60910 data85.2,-2.0,ori3,15
60920 data84.1,-1.2,ori4,15
60930 data83.0,-0.3,ori5,15
60940 data81.3, 6.4,bellatrix,15
60950 data78.6,-8.2,rigel,15
60960 data59.5,40.0,per1,16
60970 data58.5,31.9,per2,16
60980 data55.7,47.8,per3,16
60990 data51.1,49.9,algenib,16
61000 data47.0,41.0,algol,16
61010 data46.2,53.5,per6,16
61020 data31.0,42.0,alamak,17
61030 data17.4,35.6,mirach,17
61040 data 9.8,31.9,and3,17
61050 data 2.1,29.1,sirrah,17
61060 data354.8,77.6,cep1,18
61070 data342.5,66.0,cep2,18
61080 data322.2,70.6,alfrik,18
61090 data319.6,62.6,alderamin,18
61100 data269.2,51.5,dra1,19
61110 data262.6,52.3,dra2,19
61120 data231.2,59.0,dra3,19
61130 data246.0,61.5,dra4,19
61140 data257.2,65.7,dra5,19
61150 data288.1,67.7,dra6,19
61160 data31.8,23.5,hamal,20
61170 data28.7,20.8,ari2,20
61180 data84.4,21.2,tau1,21
61190 data81.6,28.6,elnath,21
61200 data69.0,16.5,aldebaran,21
61210 data58.8,12.5,tau4,21
61220 data56.9,24.1,plejaden,21
61230 data114.8,5.2,prokyon,22
61240 data111.8,8.3,cmi2,22
62000 rem ** daten sternbilder **
62010 dataursa minor,kleiner wagen,6,1
62020 dataursa maior,grosser wagen,6,8
62030 datacassiopeia,kassiopeia,4,15
62040 datapegasus,pegasus,2,20
62050 datacygnus,schwan,4,23
62060 dataaquilla,adler,5,28
62070 datalyra,leier,4,34
62080 datascorpius,skorpion,9,39
62090 databootes,bootes,5,49
62100 datavirgio,jungfrau,6,55
62110 datacrux,kreuz sueden,3,62
62120 dataleo,loewe,6,66
62130 datagemini,zwillinge,5,73
62140 datacanis maior,grosser hund,4,79
62150 dataauriga,fuhrmann,4,84
62160 dataorion,orion,6,89
62170 dataperseus,perseus,5,96
62180 dataandromeda,andromeda,3,102
62190 datacepheus,kepheus,3,106
62200 datadraco,drache,5,110
62210 dataaries,widder,1,116
62220 datataurus,stier,4,118
62230 datacanis minor,kleiner hund,1,123
62500 rem ** daten planeten **
62510 datamerkur,4.0923, 31.19, 76.987
62520 data23.00,.2056, 47.826,7.004, 0.3
871
62530 datavenus ,1.6021, 80.85,131.149
62540 data 0.76,.0068, 76.410,3.394, 0.7
233
62550 datamars  ,0.5240,144.14,335.507
62560 data11.00,.0934, 49.326,1.850, 1.5
237
62570 datajupiter,.0831,316.19, 13.839
62580 data 5.30,.0485,100.146,1.305, 5.2
028
62590 datasaturn,0.0335,158.36, 92.460
62600 data 5.50,.0557,113.511,2.486, 9.5
810
62610 datauranus,0.0117, 98.38,170.173
62620 data 5.70,.0472, 73.847,0.773,19.1
823
62700 rem ** e n d e **
```

Listing Sternenhimmel (Schluß)

Trickfilm mit dem C 64

In die vierte Dimension, die bewegte dreidimensionale Grafik, dringen Sie mit unserem Listing des Monats vor.

Der »3D-Movie Maker« muß in zwei Teilen eingegeben werden: zuerst das Maschinenprogramm, dann der Basic-Teil. Wenn das Maschinenprogramm eingegeben und gestartet wurde, so speichert es sich, falls kein Prüfsummenfehler auftrat, selbst als »TRICK.OBJ« ab. Man braucht sich nun nicht mehr darum zu kümmern. Bei dem Basic-Teil, der nun eingegeben werden kann, dürfen alle REM-Zeilen ersatzlos wegfallen. Beim Start des Basic-Teils wird automatisch der Maschinenteil, der sich auf Diskette befinden sollte, nachgeladen.

Bedienung — Eingabe des Körpers

Vor der Benutzung des Programmes, müssen die Punkte Verbindungs- und Bewegungsvorschrift des Körpers in den DATA-Zeilen ab 8000 festgelegt werden. Dies ist notwendig, damit bei Änderungen oder einem erneuten Start nicht alles wieder neu eingegeben werden muß. Die DATA-Zeilen, die das Listing momentan enthält, erzeugen den auf dem Bildschirm herumfliegenden Schriftzug »64'er«. Die Punkte werden mit X, Y und Z-Koordinaten eingegeben. Als Endmarke dient hier dreimal die 1000. Bei der Verbindungsvorschrift wird jeweils der Anfangs- und Endpunkt angegeben zum Beispiel: Von Punkt 1 nach Punkt 2 = DATA 1,2. Hier dient zweimal die 1000 als Endmarkierung.

Beispiel: Man will ein Kreuz erzeugen.
Punkte: DATA 0,10,0,10,0,0,—10,0
DATA —10, 0,0, 1000,1000,1000
Verbindungsvorschrift: DATA 1,3,2,4,1000,1000

Bei der Bewegungsvorschrift ist es allerdings etwas komplizierter. Hier müssen zweimal drei Verschiebungsarten, drei Drehungsarten und die Dauer des Vorgangs angegeben werden. Zuerst kommt die erste Verschiebung in X-, Y- und Z-Richtung, dann die Drehung um die drei Achsen, nun die zweite Verschiebung und zum Schluß die Dauer des Ganzen. Diese Reihenfolge hat folgenden Sinn:

Wird zuerst verschoben und dann gedreht, dreht sich die Figur um den Bildschirmmittelpunkt. Bei umgekehrter Reihenfolge dreht sich die Figur an beliebiger Stelle um ihren eigenen Mittelpunkt. Hier ein Beispiel einer DATA-Zeile:
DATA 0,0,0,5,0,0, 0,5,0, 10

Die Figur bewegt sich 10 Bilder lang um jeweils 5 Stellen nach oben und dreht sich dabei um jeweils fünf Grad um die X-Achse. Einen Zoomeffekt erreicht man durch Verschieben in der Z-Achse, so kann man zum Beispiel mit DATA 0,0,0,0,0,0,0,0,1,50 eine Figur langsam vergrößern. Hier bildet einmal die 1000 die Schlußmarkierung. Um die vielen verschiedenen Möglichkeiten der Bewegung zu entdecken, lohnt es sich, die verschiedenen Kombinationen auszuprobieren (und dabei mit einfachen Bewegungen anzufangen).

Bedienung — Ablauf des Programms

Der »3D-Movie-Maker« stellt vier Menüpunkte zur Wahl:
1. Erzeugen einer Grafik

Hierbei berechnet der Computer die Anzahl der Bilder und fragt diese noch einmal ab. Sollen alle Bilder gezeichnet werden, braucht nur RETURN eingegeben werden. Nun wird der Speicherbedarf berechnet; reicht der vorhandene Speicher von 23 KByte aus, so beginnt der Rechenvorgang. Dabei wird die Anzahl der fertigen Bilder angezeigt. Ist der Rechenvorgang beendet (bei 150 Bildern mit je 10 Punkten zirka eine halbe Stunde) kehrt das Programm ins Menü zurück. Durch das Drücken der Leertaste kann man sich während der Berechnungen den Film ansehen und dann durch nochmaliges Betätigen fortfahren.

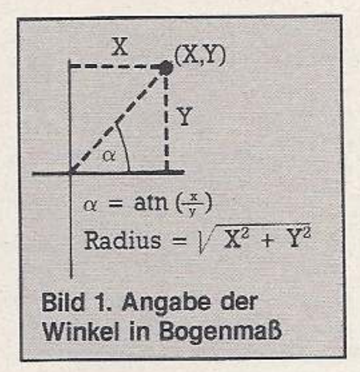


Bild 1. Angabe der Winkel in Bogenmaß

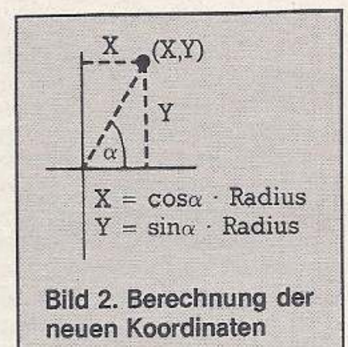


Bild 2. Berechnung der neuen Koordinaten

Variablenliste

Felder:	
X(I), Y(I), Z(I)	: Originalkoordinaten der Figur aus den DATA-Zeilen
X1(I), Y1(I), Z(I)	: Koordinaten beim Rechnen und für das POKEN
P1(I), P2(I)	: Verbindungsvorschriften aus den DATA-Zeilen
Normale Variablen:	
PO	: Speicherpointer für fertige Daten
Q-1	: Zähler für fertige Bilder
A2	: Anzahl der Linien pro Bild
A1	: Anzahl der Punkte pro Bild
AN	: Anzahl der Bilder
P1, P2, P3, P4, PB, PA	: Variablen zur Berechnung des zu löschenden Bildschirmbereichs
WX, WY, WZ, XA, YA, ZA, XB, YB, ZB	: Variablen für die Drehbewegungen und Verschieben (werden bei jedem Bild auf W1, ... aufaddiert)
W1, W2, W3, X1, Y1, Z1, X2, Y2, Z2	: Momentane Lage des Körpers

2./3. Laden/ Speichern

Da das Erstellen eines Films recht lange dauert, kann man fertige Filme abspeichern und laden. Das Programm hängt an den Filenamen automatisch ein ».GRA« an, so daß die Dateien in der Directory sofort erkennbar sind. Tritt ein Disk-Error beim Laden oder Speichern auf, landet man wieder in der »INPUT FILENAME«-Zeile. Hier noch ein Hinweis zur LOAD-Routine: Da das Einlesen von 23 KByte mit einer GET-Schleife nicht einwandfrei funktionierte und 15 Minuten dauerte, verwendet die jetzige Version die LOAD-Routine des Betriebssystems.

4. Abspielen

Beim Aufruf dieser Funktion wird man zuerst nach der Anzahl der Durchläufe (maximal 255) und nach der Anzahl der Bilder je Lauf gefragt. Beim letzteren braucht man nur RETURN eingeben, um alle Bilder ablaufen zu lassen.

Der Algorithmus

Das eigentliche Kernstück des Programms ist ein Algorithmus, der die Punkte um einen Winkel dreht, die X-, Y- und Z-Koordinaten in Bildschirmkoordinaten umrechnet und dann die außerhalb liegenden Teile der Linien wegstreicht. Beim Drehen (hier um die Z-Achse) werden erst X- und Y-Koordinaten in Radius und Winkel (im Bogenmaß) umgerechnet; bezogen auf den Koordinatenursprung (siehe Bild 1). Nun wird der Winkel, um den gedreht werden soll, addiert und die neuen Koordinaten werden aus dem neuen Winkel und dem Radius errechnet (siehe Bild 2).

Beim Umrechnen in Bildschirmkoordinaten wird einfach die Z-Koordinate auf die X und Y-Koordinate so aufgerechnet, daß Punkte, die weiter vorne liegen, vom Mittelpunkt wegrücken. Außerdem wird der Koordinatenursprung in die Bildschirmmit-

te verlegt. Durch dieses Verfahren wird die Figur mit einem Fluchtpunkt in der Mitte des Bildschirms dargestellt. Das nun folgende Wegstreichen der außerhalb liegenden Linien und Linienstücke geschieht mit Hilfe einer Gradengleichung. Falls nur Teile der Geraden außerhalb des Bildschirms liegen, wird der äußerste Wert für X beziehungsweise Y angenommen und die fehlende Koordinate errechnet (siehe Bild 3).

Beschreibung der Maschinenroutine

Die Zeichenroutine des »3D-Movie-Makers« übernimmt die Verwaltung der Grafik und ist, zum Erreichen eines Trickfilmeffekts, ganz auf Geschwindigkeit ausgelegt. Aus diesem Grund erhielt sie die folgenden Merkmale:

— Sie ist in Assembler geschrieben.
— Sie berechnet keine Punkte, sondern zeichnet nur.
— Die Koordinaten werden nicht geprüft, das heißt: das Basic-Programm darf keine »unmöglichen« Koordinaten übermitteln.
— Der Bildschirm wird nur so weit gelöscht, wie es nötig ist.
— Die Zeropage-Addressierung wird in breitem Umfang benutzt.
— Der IRQ wird abgestellt.

Das wichtigste Mittel zur Erzeugung eines flüssigen Bilderablaufs und die Grundidee der Routine ist jedoch das »verdeckte Zeichnen«. Dafür werden zwei Bitmaps benötigt (bei unserem Programm ab $A000 und $E000). Während nun eine der beiden zu sehen ist, wird auf der anderen gezeichnet. Nun wird das neue Bild sichtbar gemacht und das alte, welches nicht mehr zu sehen ist, wird gelöscht. Dieser Vorgang wiederholt sich, bis alle Bilder abgearbeitet sind. Hier noch eine Anmerkung: die Lineroutine haben wir mit einigen Änderungen dem Artikel »Ein schneller Drawline-Algorithmus« aus dem 64'er, 4/84 entnommen. Die Funktion läßt sich am besten dort nachvollziehen. (Armin und Dirk Biernaczyk/rg)

```
PROGRAMM : TRICK.OBJ          C000 C2A3
---------------------------------------

C000 : 20 36 C0 20 A4 C0 20 98   71
C008 : C0 20 E6 C0 C6 4F D0 F3   BC
C010 : 20 7F C0 C6 52 D0 EC A9   AB
C018 : 37 85 01 58 A9 00 85 C6   9B
C020 : A5 C6 F0 FC C6 C6 A9 1B   83
C028 : 8D 11 D0 A9 15 8D 18 D0   67
C030 : A9 03 8D 00 DD 60 20 FD   1C
C038 : AE 20 9E B7 86 4E 20 FD   EC
C040 : AE 20 9E B7 86 50 20 FD   04
C048 : AE 20 9E B7 86 52 A9 3B   BD
C050 : 8D 11 D0 A9 38 8D 18 D0   C1
C058 : A9 00 8D 00 DD 20 C2 C0   D0
C060 : 78 A9 35 85 01 A2 0B BD   78
C068 : 8C C0 95 A0 CA 10 F8 A2   24
C070 : 20 A9 00 A8 20 A0 00 A9   D4
C078 : A0 85 FC A9 00 85 53 A9   1C
C080 : 8B 85 8E A9 FA 85 8D A5   04
C088 : 4E 85 4F 60 99 00 E0 C8   27
C090 : D0 FA E6 A2 CA D0 F5 60   B7
C098 : A5 50 85 51 20 70 C1 C6   0B
C0A0 : 51 D0 F9 60 A0 05 B1 8D   F8
C0A8 : AA 88 B1 8D 05 FC 85 A2   48
C0B0 : 38 A5 8D E9 02 85 8D B0   3F
C0B8 : 02 C6 8E A9 00 A8 20 A0   FD
C0C0 : 00 60 A9 10 A0 CC 84 8C   F8
C0C8 : A0 00 84 8B 20 D9 C0 A0   10
C0D0 : 8C 84 8C A0 00 20 D9 C0   BF
C0D8 : 60 A2 04 91 8B C8 D0 FB   F7
C0E0 : E6 8C CA D0 F6 60 A5 53   89
C0E8 : D0 0C E6 53 A9 E0 85 FC   94
C0F0 : A5 53 8D 00 DD 60 C6 53   45
C0F8 : A9 A0 4C EE C0 8A 4A 4A   00
C100 : 29 FE A8 B9 36 C1 85 FD   8D
C108 : B9 37 C1 85 FE 8A 29 07   75
C110 : 18 65 FD 85 FD A5 14 29   BA
C118 : F8 65 FD 85 FD A5 FE 65   C7
C120 : FC 65 15 85 FE A5 14 29   85
C128 : 07 49 07 AA BD 68 C1 A0   52
C130 : 00 11 FD 91 FD 60 00 00   4D
C138 : 40 01 80 02 C0 03 00 05   87
C140 : 40 06 80 07 C0 08 00 0A   E5
C148 : 40 0B 80 0C C0 0D 00 0F   42
C150 : 40 10 80 11 C0 12 00 14   9F
C158 : 40 15 80 16 C0 17 00 19   FD
C160 : 40 1A 80 1B C0 1C 00 1E   5A
C168 : 01 02 04 08 10 20 40 80   71
C170 : 20 71 C2 A0 01 84 62 84   D4
C178 : 5F 84 5D 88 84 5E 84 61   92
C180 : 84 60 88 A5 59 C5 15 90   44
C188 : 08 D0 18 A5 58 C5 14 B0   18
C190 : 12 38 A5 14 E5 58 85 5B   98
C198 : A5 15 E5 59 85 5C 84 62   7E
C1A0 : 4C B0 C1 38 A5 58 E5 14   99
C1A8 : 85 5B A5 59 E5 15 85 5C   45
C1B0 : A5 FF C5 57 B0 0C 38 A5   49
C1B8 : 57 E5 FF 85 5A 84 5F 4C   92
C1C0 : C6 C1 E5 57 85 5A A5 5C   46
C1C8 : D0 19 A5 5B C5 5A B0 13   12
C1D0 : A6 5A 85 5A 86 5B A5 62   EF
C1D8 : 85 60 A3 5F 85 61 C8 84   72
C1E0 : 62 84 5F A5 5C 4A 85 59   F2
C1E8 : A5 5B 6A 85 58 4C 5C C2   65
C1F0 : A5 62 30 0B 18 65 14 85   3C
C1F8 : 14 90 0D E6 15 D0 09 18   A1
C200 : 65 14 85 14 B0 02 C6 15   B4
C208 : 18 A5 57 65 61 85 57 18   45
C210 : A5 58 65 5A 85 58 A5 59   EA
C218 : 69 00 85 59 E6 5D D0 02   AE
C220 : E6 5E A5 59 C5 5C 90 34   B4
C228 : D0 06 A5 5B C5 58 B0 2C   0A
C230 : 38 A5 58 E5 5B 85 58 A5   9C
C238 : 59 E5 5C 85 59 A5 60 30   F0
C240 : 0B 18 65 14 85 14 90 0D   89
C248 : E6 15 D0 09 18 65 14 85   16
C250 : 14 B0 02 C6 15 18 A5 57   6D
C258 : 65 5F 85 57 A6 57 20 FD   5B
C260 : C0 A5 5E C5 5C 90 07 A5   F5
C268 : 5B C5 5D B0 01 60 4C F0   39
C270 : C1 A0 05 B1 8D 85 14 88   5F
C278 : B1 8D 10 05 68 68 4C 97   BF
C280 : C2 85 15 88 B1 8D 85 57   A7
C288 : 88 B1 8D 85 58 88 B1 8D   A8
C290 : 85 59 88 B1 8D 85 FF 38   90
C298 : A5 8D E9 06 85 8D B0 02   CA
C2A0 : C6 8E 60 DF 5E DE 5E DE   D5
```

Listing »TRICK.OBJ«. Verwenden Sie bitte zur Eingabe den MSE

```
100 REM ------------------------------          <012>
105 REM -- 3D-MOVIE-MAKER                 --    <109>
110 REM -- EIN PROGRAMM VON:              --    <031>
115 REM -- DIRK & ARMIN BIERNACZYK        --    <111>
120 REM -- AN DER PAPENBURG 41            --    <046>
125 REM -- 4630 BOCHUM 6                  --    <129>
130 REM -- TEL.: //////                   --    <044>
135 REM ------------------------------          <047>
140 REM 1985 BY ARMIN & DIRK                    <232>
145 REM             BIERNACZYK                  <018>
150 :                                           <208>
160 REM ----------------                        <089>
170 REM --- HAUPTMENUE ---                      <067>
180 REM ----------------                        <109>
190 :                                           <248>
191 IF A=0 THEN A=1:LOAD"TRICK.OBJ",8,1         <144>
192 :                                           <250>
195 POKE 56,50:CLR:REM SPEICHER HERAB.          <216>
200 PRINT CHR$(147)                             <021>
210 PRINT SPC(10)"** 3D-MOVIE-MAKER **"         <072>
220 PRINT:PRINT:PRINT:PRINT                     <238>
230 PRINT SPC(9)"1 - GRAFIK ERSTELLEN"          <139>
235 PRINT                                       <132>
240 PRINT SPC(9)"2 - GRAFIK ABSPIELEN"          <123>
245 PRINT                                       <142>
250 PRINT SPC(9)"3 - GRAFIK LADEN"              <087>
255 PRINT                                       <152>
260 PRINT SPC(9)"4 - GRAFIK ABSPEICHERN"        <035>
262 PRINT                                       <160>
264 PRINT SPC(9)"5 - ENDE"                      <108>
265 :                                           <068>
```

Listing »3D-Movie-Maker«. Beachten Sie bitte bei der Eingabe den Checksummer 64.

```
270 GET W$:IF W$<"1"OR W$>"5"THEN 270          <040>
280 W=VAL(W$)                                  <179>
285 IF W=5 THEN END                            <014>
290 ON W GOSUB 1040,5040,6040,7040             <063>
300 GOTO 200                                   <072>
310 :                                          <113>
315 :                                          <118>
320 :                                          <123>
1000 REM -----------------                     <119>
1010 REM --- ERSTELLEN ---                     <064>
1020 REM -----------------                     <139>
1030 :                                         <068>
1040 GOSUB 4540    :REM VARIABLEN              <203>
1050 GOSUB 4040    :REM EINLESEN               <143>
1060 PRINT CHR$(147)"FERTIGE BILDER: "         <171>
1061 IF AN*(A2*6+2)<23000 THEN 1080            <137>
1062 PRINT"ZU WENIG SPEICHERPLATZ"             <014>
1063 POKE 198,0:WAIT 198,1:RETURN              <083>
1070 :                                         <108>
1080 FOR Q=1 TO AN  :REM ANZAHL DER BILDER     <056>
1085 GOSUB 3040    :REM BEWEGEN                <091>
1090 GOSUB 2040    :REM BERECHNEN              <236>
1095 GET TA$:IF TA$=" "THEN IF Q>1 THEN SY
     S 49152,Q-1,A2,1                          <029>
1100 NEXT                                      <210>
1110 Q=Q-1                                     <138>
1120 RETURN                                    <242>
1130 :                                         <168>
1135 :                                         <173>
2000 REM -----------------                     <099>
2010 REM --- BERECHNEN ---                     <008>
2020 REM -----------------                     <119>
2030 :                                         <047>
2040 REM --- VERSCHIEBEN1 ---                  <251>
2050 :                                         <068>
2060 FOR I=1 TO A1                             <21[illegible]
2070 X1(I)=X(I)+X1:Y1(I)=Y(I)+Y1               <079>
2080 Z1(I)=Z(I)+Z1                             <008>
2090 NEXT                                      <180>
2100 :                                         <118>
2110 REM --- DREHEN ---                        <153>
2120 :                                         <138>
2130 IF W1=0 THEN 2250                         <191>
2140 FOR I=1 TO A1                             <039>
2150 XD=X1(I):YD=Y1(I)                         <140>
2160 IF XD=0 THEN XD=1E-20                     <249>
2165 IF YD=0 THEN YD=1E-20                     <000>
2170 R=SQR(XD*XD+YD*YD)                        <005>
2180 W=ATN(YD/XD)                              <141>
2190 IF XD>0 AND YD<0 THEN W=W+π*2:GOTO 22
     10                                        <227>
2200 IF XD<0 THEN W=W+π                        <090>
2210 W=W+W1                                    <060>
2220 Y1(I)=SIN(W)*R:X1(I)=COS(W)*R             <098>
2230 NEXT                                      <064>
2240 :                                         <002>
2250 IF W2=0 THEN 2370                         <059>
2260 FOR I=1 TO A1                             <159>
2270 ZD=Z1(I):YD=Y1(I)                         <008>
2280 IF ZD=0 THEN ZD=1E-20                     <117>
2285 IF YD=0 THEN YD=1E-20                     <120>
2290 R=SQR(ZD*ZD+YD*YD)                        <129>
2300 W=ATN(YD/ZD)                              <007>
2310 IF ZD>0 AND YD<0 THEN W=W+π*2:GOTO 23
     30                                        <097>
2320 IF ZD<0 THEN W=W+π                        <213>
2330 W=W+W2                                    <182>
2340 Y1(I)=SIN(W)*R:Z1(I)=COS(W)*R             <221>
2350 NEXT                                      <185>
2360 :                                         <123>
2370 IF W3=0 THEN 2510                         <177>
2380 FOR I=1 TO A1                             <024>
2390 ZD=Z1(I):XD=X1(I)                         <127>
2400 IF ZD=0 THEN ZD=1E-20                     <238>
2405 IF XD=0 THEN XD=1E-20                     <239>
2410 R=SQR(ZD*ZD+XD*XD)                        <248>
2420 W=ATN(XD/ZD)                              <127>
2430 IF ZD>0 AND XD<0 THEN W=W+π*2:GOTO 24
     50                                        <219>
2440 IF ZD<0 THEN W=W+π                        <077>
2450 W=W+W3                                    <047>
2460 X1(I)=SIN(W)*R:Z1(I)=COS(W)*R             <084>
2470 NEXT                                      <049>
2480 :                                         <243>
2490 REM --- UMRECHNEN ---                     <005>
2500 :                                         <007>
2510 FOR I=1 TO A1                             <154>
2530 X1(I)=(X1(I)+X2)*1.01↑(Z1(I)+Z2)          <046>
2540 Y1(I)=(Y1(I)+Y2)*1.01↑(Z1(I)+Z2)          <059>
2560 NEXT                                      <140>
2565 REM 1.01 KANN LEICHT GEAENDERT            <206>
2567 REM WERDEN                                <101>
2580 :                                         <088>
2590 PA=0:PB=199                               <188>
2600 PO=PO-2                                   <255>
2601 REM -----------------------------         <219>
2602 REM --- UBERGETRETENE LINIEN  ---         <097>
2603 REM --- BERECHNEN UND POKEN   ---         <192>
2604 REM -----------------------------         <222>
2610 FOR I=1 TO A2                             <000>
2620 X0%=X1(P1(I)):Y1%=Y1(P1(I))               <043>
2630 X2%=X1(P2(I)):Y2%=Y1(P2(I))               <058>
2640 X1%=0:X3%=0:ME=0                          <100>
2641 IF X0%> 159 AND X2%> 159 THEN 2760        <007>
2642 IF X0%<-159 AND X2%<-159 THEN 2760        <098>
2643 IF Y1%<- 99 AND Y2%<- 99 THEN 2760        <012>
2644 IF Y1%>  99 AND Y2%>  99 THEN 2760        <179>
2650 IF Y1%=Y2%THEN 2711                       <114>
2655 IF X2%=X0%THEN 2690                       <122>
2660 M=(Y2%-Y1%)/(X2%-X0%)                     <205>
2670 B=-M*X0%+Y1%                              <022>
2680 GOTO 2720                                 <214>
2690 IF Y1%>99 OR Y1%<-99 THEN Y1%=99*SGN(
     Y1%)                                      <242>
2700 IF Y2%>99 OR Y2%<-99 THEN Y2%=99*SGN(
     Y2%)                                      <000>
2710 GOTO 2760                                 <248>
2711 IF X0%>159 OR X0%<-159 THEN X0%=159*S
     GN(X0%)                                   <134>
2712 IF X2%>159 OR X2%<-159 THEN X2%=159*S
     GN(X2%)                                   <143>
2713 GOTO 2760                                 <251>
2720 IF X0%>159 OR X0%<-159 THEN X0%=159*S
     GN(X0%):Y1%=M*X0%+B                       <188>
2730 IF X2%>159 OR X2%<-159 THEN X2%=159*S
     GN(X2%):Y2%=M*X2%+B                       <209>
2740 IF Y1%>99 OR Y1%<-99 THEN Y1%=99*SGN(
     Y1%):X0%=(Y1%-B)/M                        <164>
2750 IF Y2%>99 OR Y2%<-99 THEN Y2%=99*SGN(
     Y2%):X2%=(Y2%-B)/M                        <181>
2760 IF Y1%>99 OR Y1%<-99 THEN X1%=255:X0%
     =0:Y1%=0:Y2%=0:X2%=0:GOTO 2810            <186>
2770 IF X0%>159 OR X0%<-159 THEN X1%=255:X
     0%=0:Y1%=0:Y2%=0:X2%=0:GOTO 2810          <026>
2780 :                                         <032>
2790 X0%=X0%+160:X2%=X2%+160                   <200>
2791 Y1%=Y1%+100:Y2%=Y2%+100                   <195>
2793 IF Y1%>PA THEN PA=Y1%                     <008>
2794 IF Y2%>PA THEN PA=Y2%                     <011>
2795 IF Y1%<PB THEN PB=Y1%                     <014>
2796 IF Y2%<PB THEN PB=Y2%                     <017>
2799 IF X0%>255 THEN X0%=X0%-256:X1%=1         <068>
2800 IF X2%>255 THEN X2%=X2%-256:X3%=1         <077>
2805 :                                         <057>
2810 POKE PO,X0%:POKE PO-1,X1%                 <057>
2820 POKE PO-2,Y1%:POKE PO-3,X2%               <038>
2830 POKE PO-4,X3%:POKE PO-5,Y2%               <054>
2840 PO=PO-6                                   <244>
2850 NEXT                                      <175>
2860 :                                         <113>
2862 IF PA<PB THEN PA=199:PB=0                 <213>
2863 PA=40*((PA OR 7)+1)/256+1                 <167>
2864 PB=40*(PB AND 248)/256                    <009>
2866 POKE PO+6*A2+2,INT(P1)-INT(P3)            <053>
2868 POKE PO+6*A2+1,P3                         <254>
2870 P1=P2:P2=PA:P3=P4:P4=PB                   <234>
2872 PRINT CHR$(19)SPC(16)Q                    <229>
2880 RETURN                                    <217>
2890 :                                         <143>
2900 :                                         <153>
3000 REM ---------------                       <245>
3010 REM --- BEWEGEN ---                       <103>
3020 REM ---------------                       <009>
3030 :                                         <027>
3040 IF E=0 THEN 3100                          <008>
3050 W1=W1+WZ:W2=W2+WX:W3=W3+WY                <195>
3060 X1=X1+XA:Y1=Y1+YA:Z1=Z1+ZA                <145>
3070 X2=X2+XB:Y2=Y2+YB:Z2=Z2+ZB                <164>
3080 E=E-1:RETURN                              <244>
3090 :                                         <088>
3100 READ XA,YA,ZA,WX,WY,WZ,XB,YB,ZB,E         <047>
3110 WX=WX*π/180:WY=WY*π/180                   <112>
```

```
3120 WZ=WZ*π/180:YA=-YA:YB=-YB              <217>
3130 GOTO 3050                              <151>
3140 :                                      <138>
3150 :                                      <148>
4000 REM ----------------                   <014>
4010 REM --- EINLESEN ---                   <169>
4020 REM ----------------                   <034>
4030 :                                      <007>
4040 I=0                                    <002>
4050 I=I+1                                  <000>
4060 READ X(I),Y(I),Z(I)                    <163>
4070 Y(I)=-Y(I)                             <056>
4080 IF X(I)<1000 THEN 4050                 <096>
4090 A1=I-1                                 <082>
4100 I=0                                    <063>
4110 I=I+1                                  <061>
4120 READ P1(I),P2(I)                       <018>
4130 IF P1(I)<1000 THEN 4110                <185>
4140 A2=I-1                                 <134>
4150 :                                      <128>
4155 AN=0                                   <188>
4157 READ A:IF A=1000 THEN 4180             <002>
4160 FOR I=1 TO 9:READ A:NEXT               <152>
4170 AN=AN+A:GOTO 4157                      <169>
4180 RESTORE                                <240>
4190 READ A,A,A                             <016>
4200 IF A<1000 THEN 4190                    <045>
4210 READ A,A                               <183>
4220 IF A<1000 THEN 4210                    <058>
4230 :                                      <208>
4240 PRINT CHR$(147)"BILDERZAHL{2SPACE}"AN  <161>
4250 PRINT CHR$(19)SPC(11);:INPUT I         <057>
4260 IF I>255 OR I>AN THEN 4240             <127>
4270 AN=I                                   <072>
4280 RETURN                                 <086>
4290 :                                      <012>
4300 :                                      <022>
4500 REM -----------------                  <049>
4510 REM --- VARIABLEN ---                  <224>
4520 REM -----------------                  <069>
4530 :                                      <253>
4533 REM NACH BEDARF DIMENSIONIEREN         <044>
4536 :                                      <003>
4540 DIM X(50),Y(50),Z(50)                  <216>
4560 DIM X1(50),Y1(50),Z1(50)               <127>
4570 DIM P1(50),P2(50)                      <012>
4580 P0=35839:P1=32:P2=32:P3=0:P4=0         <054>
4590 RETURN                                 <141>
4600 :                                      <067>
4605 :                                      <072>
5000 REM -----------------                  <039>
5010 REM --- ABSPIELEN ---                  <213>
5020 REM -----------------                  <059>
5030 :                                      <243>
5040 IF Q>0 AND A2>0 THEN 5090              <248>
5050 PRINT CHR$(147):PRINT:PRINT:PRINT      <147>
5060 PRINT SPC(5)"ES GIBT KEINE GRAFIK"     <150>
5070 POKE 198,0:WAIT 198,1:POKE 198,0       <016>
5080 RETURN                                 <121>
5090 INPUT"{CLR}WIEVILE DURCHLAUEFE";DU     <098>
5091 IF DU>255 OR DU<1 THEN 5090            <009>
5092 PRINT"{HOME,2DOWN}WIEVIELE BILDER{5SP
     ACE}"Q                                 <102>
5093 PRINT"{UP}"SPC(19);:INPUT I            <226>
5094 IF I>Q OR I<1 THEN 5092                <035>
5100 :                                      <057>
5105 SYS 49152,I,A2,DU:RETURN               <072>
5110 :                                      <067>
6000 REM -------------                      <095>
6010 REM --- LADEN ---                      <146>
6020 REM -------------                      <115>
6030 :                                      <223>
6040 PRINT CHR$(147):PRINT:PRINT:PRINT:PRI
     NT                                     <072>
6050 INPUT"{3SPACE}FILENAME: ";NA$          <235>
6055 IF NA$="M"THEN RETURN                  <116>
6060 OPEN 2,8,2,NA$+".GRA,S,R"              <040>
6070 OPEN 1,8,15:INPUT#1,FE$                <093>
6080 IF FE$="00"THEN 6090                   <221>
6082 CLOSE 1:CLOSE 2:GOTO 6040              <067>
6090 GET#2,Q$,A2$                           <059>
6100 Q=ASC(Q$):A2=ASC(A2$)                  <135>
6110 AD=35839-Q*(A2*6+2)                    <098>
6140 CLOSE 2:CLOSE 1                        <240>
6150 AH=INT(AD/256):AL=AD-AH*256            <169>
6160 A$=NA$+".GRA,S"                        <103>
6170 FOR I=51000 TO 51000+LEN(A$)-1         <061>
6180 POKE I,ASC(MID$(A$,I-50999,1))         <108>
6190 NEXT                                   <200>
6200 POKE 183,LEN(A$)                       <040>
6210 POKE 187,56:POKE 188,199               <105>
6220 POKE 185,0:POKE 186,8:POKE 147,0       <146>
6570 POKE 195,AL:POKE 196,AH                <216>
6580 SYS 62648                              <117>
6590 RETURN                                 <101>
6600 :                                      <027>
6610 :                                      <037>
6620 :                                      <047>
7000 REM -------------------                <089>
7010 REM --- ABSPEICHERN ---                <062>
7020 REM -------------------                <109>
7030 :                                      <203>
7040 IF Q>0 AND A2>0 THEN 7090              <210>
7050 PRINT CHR$(147):PRINT:PRINT:PRINT      <107>
7060 PRINT SPC(5)"ES GIBT KEINE GRAFIK"     <110>
7070 POKE 198,0:WAIT 198,1:POKE 198,0       <232>
7080 RETURN                                 <081>
7090 PRINT CHR$(147):PRINT:PRINT:PRINT      <147>
7100 INPUT"{4SPACE}FILENAME: ";NA$          <009>
7105 IF NA$="M"THEN RETURN                  <146>
7110 OPEN 2,8,2,NA$+".GRA,S,W"              <075>
7120 OPEN 1,8,15:INPUT#1,FE$                <123>
7130 IF FE$="00"THEN 7140                   <248>
7135 CLOSE 1:CLOSE 2:GOTO 7090              <106>
7140 PRINT#2,CHR$(Q);CHR$(A2);              <095>
7150 FOR I=35839-Q*(A2*6+2)TO 35839         <107>
7160 PRINT#2,CHR$(PEEK(I));:NEXT            <116>
7170 CLOSE 2:CLOSE 1:RETURN                 <195>
7175 :                                      <093>
7180 :                                      <098>
7185 :                                      <103>
8000 REM --------------------------         <083>
8010 REM --- DATAS FUER PUNKTE ---          <124>
8020 REM --------------------------         <103>
8030 :                                      <183>
8040 DATA -20, 10,0                         <130>
8050 DATA -30, 10,0                         <141>
8060 DATA -30,-10,0                         <196>
8070 DATA -20,-10,0                         <205>
8080 DATA -20,  0,0                         <121>
8090 DATA -30,  0,0                         <132>
8100 DATA -15, 10,0                         <194>
8110 DATA -15,  0,0                         <155>
8120 DATA - 5, 10,0                         <165>
8130 DATA - 5,  0,0                         <126>
8140 DATA - 5,-10,0                         <230>
8150 DATA  10, 10,0                         <194>
8160 DATA   5,  5,0                         <116>
8170 DATA  15,-10,0                         <008>
8180 DATA   5,-10,0                         <225>
8190 DATA   5,  0,0                         <141>
8200 DATA  15,  0,0                         <169>
8210 DATA  15, -5,0                         <229>
8220 DATA   5, -5,0                         <190>
8230 DATA  20,-10,0                         <033>
8240 DATA  20,  0,0                         <205>
8250 DATA  20, -5,0                         <009>
8260 DATA  30,  0,0                         <226>
8900 DATA 1000,1000,1000                    <004>
8910 :                                      <042>
9000 REM -----------------------------      <198>
9010 REM --- VERBINDUNGSVOSCHRIFT ---       <241>
9020 REM -----------------------------      <218>
9030 :                                      <163>
9040 DATA  1, 2,  2, 3,  3, 4,  4, 5        <194>
9050 DATA  5, 6,  7, 8,  8,10,  9,11        <066>
9060 DATA 12,13, 14,15, 15,16, 16,17        <108>
9070 DATA 17,18, 18,19, 20,21, 22,23        <122>
9450 DATA 1000,1000                         <063>
9460 :                                      <082>
9500 REM ---------------------------        <098>
9510 REM --- BEWEGUNGSVOSCHRIFT ---         <071>
9520 REM ---------------------------        <118>
9530 :                                      <153>
9531 REM XA,YA,ZA,WX,WY,WZ,XB,YB,ZB,E       <111>
9550 DATA 0,0,  1 ,0,0,0 ,0,0,  0 ,30       <150>
9560 DATA 0,0,  0 ,0,5,0 ,0,0,  0 ,108      <218>
9570 DATA 0,0,-30 ,0,0,0 ,0,0,-30 ,1        <055>
9580 DATA 0,0,  0 ,5,0,5 ,0,0,.81 ,36       <042>
9620 DATA 1000                              <253>
```

Listing »3D-Movie-Maker« (Schluß)

Hier gibt's Clubs

Um Euch und den Clubs unnötige Kosten und Zeit zu ersparen, einige Hinweise. Bei schriftlichen Anfragen an die Clubs, ist generell ein ausreichend frankierter Umschlag beizufügen. Bevor man einem Club beitritt, sollte man sich über dessen Status, Mitgliederzahl, Beitragssätzen (soweit nicht mit aufgeführt) und Gebühren für Lehrgänge informieren lassen. Aus Platzgründen und der Übersichtlichkeit wegen sind neben den Adressen nur einige zusätzliche Schwerpunkte mit angegeben. Die Adressen sind nach Postleitzahlen geordnet.

Computerbox, Eichenallee 61, 1000 Berlin 19,
2monatlich erscheinende Computerzeitschrift, Hardware-Berichte, Softwaretests, Basic-Kurs, Berichte zu aktuellen Anlässen, Rätsel;

C 64-Club »Die Bit Teufel«, Tel. Berlin 3618491,
Erfahrungsaustausch, Spiele, Anwendersoftware, Drucker, DFÜ, Gerätebörse;

C 64 Programmer Club, Martin Paff, Hammer Str. 69, 2000 Hamburg 70,
Clubzeitschrift, Treffen, Erstellung von Software, Bibliothek;

C-64-INFO-Club L A W A, T. H. Zibell, Wakendorferweg 32, 2000 Hamburg 62, Tel. (040) 5292127

Magicclub 64, Arne Palke, Eichenbuchstr. 45, 2057 Reinbek b. Hamburg,
Clubzeitschrift, Softwarebibliothek, Treffen;

C 64-Club ACCS, Erlenweg 51, 2083 Halstenbek,
Clubzeitschrift, PRG-Bibliothek, verbilligte Disketten, Auslandskontakte, DFÜ;

HBS-Computerclub, Am Steingrab 45, 2150 Buxtehude,
Programmtausch, PRG-Bibliothek, Softwareentwicklung, eventuell Eröffnung einer Mailbox;

C 64-User-Club Lübeck, Matthias Koerner, Fregattenstr. 60, 2400 Lübeck,
Clubzeitschrift, Unterstützung beim Programmieren, Akustikkoppler, Programmtausch, Tips, Tricks;

Georg Körster, Herm-Ehlers-Str. 8, 2903 Ofen,
Clubzeitschrift, PRG-Bibliothek, Meinungs-, Erfahrungs- und Adressentausch, Lösung von Soft- und Hardwareproblemen;

SCORA-Computerclub Aurich, Manfred Janßen, Dellingstr. 10, 2965 Ihlow/Riepe
Clubtreffen, Arbeitsgruppen, Erfahrungsaustausch, Bibliothek, Softwarebibliothek, Robotersteuerung;

Sascha Troggler, Kreuzfeld 20, 3250 Hameln 1, Mailboxnr. (05151) 63687,
Clubtreffen, Erfahrungsaustausch, DFÜ, Hilfestellung bei Problemlösungen, eigene Mailbox;

Christian Hülsmann, Fliederweg 10, 3380 Goslar 1, Tel. (05321) 4898,
Softwareerstellung, bald eigene Mailbox;

Christiane Canisius, COMAL User Club, Freiheitsstr. 30, 4000 Düsseldorf 12,
COMAL-NEWS, COMAL-Literatur, Problemhilfe;

C 64-Softentwurf-Club, Cedric Volkmer, Konkordiastr. 24, 4000 Düsseldorf 1,
Softwareentwurf, Spiele, Anwender und so weiter;

Ghostbusters Dorsten, Glückaufstr. 252, 4270 Dorsten 21,
Clubtreffen, Briefkastenonkel, Knacken von Adventures, Softwarebibliothek, etc.;

64 User Club Osnabrück, Ulf Henke, Am Gretescher Turm 6, 4500 Osnabrück,
Clubtreffen (14tägig), monatliche Zeitschrift, Clubbibliothek, Softwareentwicklung (Basic, Pascal, Maschinensprache), Hardware;

Laborsoft & -Hardware, J.U.Buschmann, Schneiderstr. 80, 4600 Dortmund 50, Tel. 0231/735496
Mailbox, Clubzeitschrift, Clubtreffen, Ausflüge zu Computerveranstaltungen, gemeinsame Herstellung/Verkauf von Soft & Hardware, schnelles Erlangen von neuester Soft- und Hardware, wegen guter Auslandsverbindung, kein Clubbeitrag;

CCR (Computer-Club Reschke), Hohe Str. 21A, 4600 Dortmund 1, Tel. (0231) 160014
Commodore C 64, Clubräume, Clubbeitrag 10 Mark, Basic-Programmierkurse;

64'er Adventure Club, Christian Paas, Nordring 54, 4630 Bochum,
2monatlich erscheinende Zeitschrift, 15 Mark Jahresbeitrag, Lösen von Adventures, schriftlicher Informationsaustausch;

DATEX
Commodore Software Club
Uwe Thielker, Glückaufstr. 6, 4670 Lünen
Erfahrungsaustausch, Tips und Tricks, Hilfen für Einsteiger, spezielle Problemlösungen, evtl. Clubzeitschrift

VC 20-User Club, Ulf Kriewett, Thuermchenswall 71, 5000 Köln 1, Tel. (0221) 121634,
Clubzeitschrift, Softwarebibliothek, Hardware, Tips und Tricks;

C 64/Club Compy
Clubmagazin etc., Aufbau einer Softwarebibl.
Thomas Spieler/D., Porzerfeld 20, 5000 Köln 90;

Udo Künstler, Geilenkirchener Str. 87, 5120 Herzogenrath, Tel. (02406) 5045,
monatliches Clubtreffen, Clubzeitschrift, Beratung und Hilfestellung, Tips & Tricks, immer neue Informationen;

S.C.O.U.T.-Computerclub, Martin Mirgel, Victoriastr. 20, 5400 Koblenz,
monatliches Clubtreffen, Kontakt zu Zeitschriften, Hardware, Elektronik-Kurse für Anfänger, DFÜ, Softwarebibliothek;

C 64-Club Montabaur, Michael Bromund, Saarstr. 13, 5430 Montabaur,
Clubzeitschrift, Clubtreffen, DFÜ, Software- und Literaturbibliothek, Hardware;

Atomos-Soft, Rene Klöckner, Klagenfurtstr. 19, 5800 Hagen 7, C 64, Clubtreffen, monatl. Zeitschrift, Software, etc.;

Klaus-Dieter Friedrich, DH1FAB, Bernadottestr. 3, 6000 Frankfurt 50, Tel. (069) 571381, Mailbox OTIS (06181) 48884 Nr. 222 und 1014,
Amateurfunkthemen, Treffen, Clubzeitschrift, Informationsaustausch über das Mailboxsystem »OTIS«, Btx, Seitennummer *208247#, Jahresbeitrag von 15 Mark;

C 64-Weiterbildung, c/o Albert Pflüger, Eckenheimer Landstr. 13, 6000 Frankfurt/Main 1,
Koordination von Informationen zu C 64-Anwendungen in der Weiterbildung, unregelmäßige Mitteilungen und Treffen;

Futuresoft, Wiesenstr. 49, 6090 Rüsselsheim, Tel. (06142) 53320,
Soft- und Hardware;

Stefan Roos, Pfarrgasse 6, 6097 Trebur 3,
Treff 2mal monatlich, Beitrag 10 Mark pro Jahr, Clubzeitschrift, Software- und Bücherbibliothek;

Mark-Oliver Faßmann, Sudetenstr. 19, 6108 Weiterstadt 1,
Programmbibliothek, Modem, Clubinfo, halbjährlicher Clubbeitrag (etwa 10 Mark);

Deutsche Commodore-User, Peterswälderstr. 8, 6452 Hainburg,
Soft- und Hardwareentwicklung/-Tausch, Problemlösungen für alle Computer, monatliche Clubzeitung, Programmbibliothek, Beitrag 80 Pfennig pro Monat + fünf Programme im Jahr;

Heidelberger 64'er, Wolfgang Agostini, Karl-Metz-Str. 11, 6904 Eppelheim,
Clubtreffen, Erfahrungs- und Programmaustausch, jedes Semester wechselnder Schwerpunkt: Adventures, Assembler, kleine Hardware-Tricks;

Commodore-User-Club, Wend.-Berberich-Str. 3, 6968 Walldürn,
Infoblatt (C 64), Programmaustausch (Floppy), schriftlicher Fragenaustausch;

BNT e.V. – Öffentlichkeitsarbeit, Postfach 400846, Marktstr. 48, 7000 Stuttgart-Bad Cannstatt, Tel. (0711) 558391, (Mo-Fr: 9-16), Mailbox: (0711) 558392,
Jungendarbeit, Fortbildung, Zeitschrift, Schulungen und Kurse, Computercamps, Datenfernübertragung, Bildung von Gruppen und Treffen;

64'er Club, Michael Frieser, Hoffmannstr. 12, 7015 Korntal 1 (bei Stuttgart),
Clubtreffen, Zeitschift und Bibliothek in Vorbereitung, Hardware, Tips und Tricks, monatliche Software;

Micro Computerclub Singen, Emil Grass, Hauptstr. 34, 7709 Hilzingen, Tel. (07731) 67819,
Clubtreffen, Assemblerkurs, Erfahrungs- und PRG-Tausch, DFÜ;

Computerclub »New Line«, Postfach 8007, 7910 Neu-Ulm 8,
Speziell für Anfänger;

C 64 – ACM (e.V.), c/o Gerald Gerlich, Grasmeierstr. 25/220, 8000 München 40,
monatliche Clubzeitung, Buch- und Softwarebibliothek, Schulungen (Basic, Assembler), Hardwarebastelkreis, Sammelbestellungen, Kaufberatungen, Soft- und Hardwareentwicklungen, eigene Mailbox, Reparaturen;

CBM-64-Userclub München, Wallisersstr. 13, 8000 München 71, Tel. (089) 762424;
Treffen nach Vereinbarung, kein Beitrag, Maschinensprache, Hardware;

C 64-User Club »Flanuk«, O. Sahlmann, Irisstr. 1, 8000 München 45,
Softwaretausch, Clubzeitung (in Vorbereitung), Erfahrungsaustausch, gemeinsame Softwareproduktionen;

Marathon-Soft(ware), c/o Raimund C. Ohnemus, Daiserstr. 49, 8000 München 70,
Gemeinsames Erlernen neuer Programmiersprachen (Maschinencode, Comal, Forth, Pascal), Austausch von Kenntnissen, Gemeinsamer Softwarekauf, kein Clubbeitrag, Aufnahme von Anfängern, Umbau von Commodore-Joysticks (Joyboards für Linkshänder, Dauerfeuer);

C 64-Club PAF, CCP, Jahnhöhe 33, 8068 Pfaffenhofen 1,
Clubtreffen, Software, Clubzeitschrift;

Rüdiger Lang, Obingerstr. 4, 8201 Aindorf,
Schulung, Software, Hardware, Steuern, DFÜ, Clubtreffen;

Commodore-Club Lohr, Christian Rosenkranz, Hauptstr. 6, 8770 Lohr/Main;

Sicher sind das noch nicht alle Clubs in Deutschland. Wir bitten daher die nicht aufgeführten Clubs, sich zu melden.
In der nächsten Ausgabe sind die Clubs aus dem benachbarten Ausland an der Reihe.

Mini-Grafik VC 20

Diese kleine Grafik-Hilfe läuft auf der Grundversion des VC 20.

Wie der Name schon sagt, handelt es sich um eine Grafik-Erweiterung im kleinen Stil. Damit wird Grafik-Programmierung auch ohne große Speichererweiterung oder (teures) Grafik-Modul möglich. Da in Maschinensprache geschrieben, ist diese Basic-Erweiterung überdies sehr schnell. Minutenlange Wartezeiten auf die Fertigstellung einer Bildschirmgrafik entfallen damit.

Zur Anwendung der Grafikhilfe: Man nehme das Programm-Listing, tippe es in den VC 20 und erfreue sich folgender Befehle und Funktionen:

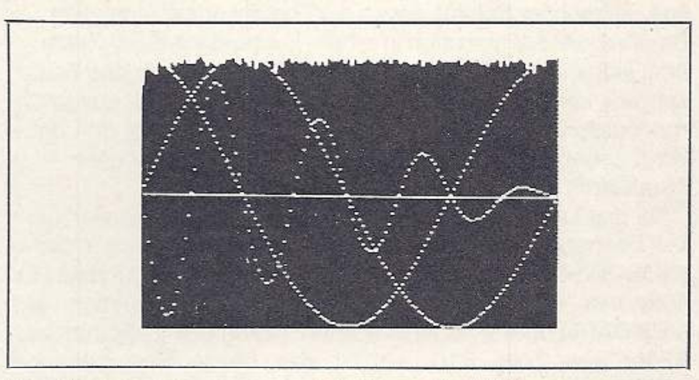
@ON	schaltet den Bildschirm auf Grafikdarstellung um, wobei die momentane Zeichenfarbe zur Plotfarbe wird
@CLR	löscht den Grafik-Speicher
@f,x,y	setzt ($f \neq 0$) oder löscht ($f=0$) einen Punkt mit der Koordinate (x,y). Für f gilt: $0 <= f < 256$, für x wie für y gilt: $0 <= x < 128$. f, x und y können beliebige Variable oder Formeln sein
@(x,y)	ist eine Funktion, welche testet, ob der Punkt mit der Koordinate (x,y) gesetzt (Ergebnis = —1) oder nicht gesetzt (Ergebnis = 0) ist. Beispiel: »PRINT @ (5,7)«
@RETURN	schaltet den Bildschirm schließlich wieder auf normalen Print-Modus

Das Programm implementiert sich nach dem Start des einzugebenden Basic-Laders (Listing 1) von selbst und ist absolut absturzsicher. Es hat, man lese und staune, eine Länge von nur 256 Byte. Für eine Auflösung von 128 mal 128 Punkten sind 2 KByte RAM unerläßlich. Somit verbleiben in der Grundversion 1279 Byte und mit 3 KByte Erweiterung sage und schreibe 4351 Byte Basic-RAM. Zugegeben, das ist nicht die Welt, aber für den Anfang durchaus ausreichend, und dafür wurde die Routine schließlich konzipiert. Überdies wird der Kassetten-Puffer nicht benutzt, und man kann von einem Programm aus mit »LOAD« ein weiteres nachladen. Listing 2 ist ein kleines Demo-Programm, um die Fähigkeiten der »Mini-Grafik« einmal vorzuführen (siehe Bild).

(Wolfgang Wirth/ev)

```
100 REM MG10 BY W.WIRTH                         <150>
105 POKE 55,0:POKE 56,21:CLR:POKE 648,22        <106>
110 SYS 58651:POKE 36879,14:PRINT"{WHITE}"      <164>
115 DEF FN B4(X)=X-48+(X>57)*7                  <065>
120 FOR I=0 TO 255:READ BY$:PRINT"{HOME}"I
    ,BY$                                        <066>
125 BY=16*FN B4(ASC(BY$))+FN B4(ASC(RIGHT$
    (BY$,1)))                                   <130>
130 CS=CS+BY:POKE 5376+I,BY:NEXT                <193>
135 FOR I=0 TO 3:READ BY:CS=CS+BY               <195>
140 POKE 6140+I,BY:NEXT                         <100>
145 IF CS=29186 THEN SYS 5389:SYS 58238         <110>
150 PRINT"{CLR,DOWN,SPACE}CHECKSUMMENFEHLE
    R !"                                        <161>
155 PRINT"{DOWN,SPACE}DATABLOCK PRUEFEN ."
    :END                                        <139>
160 DATA 19,15,83,C4,7C,C5,1A,C7,29,15,EF       <199>
165 DATA 15,00,A2,0B,BD,00,15,9D,00,03,CA       <155>
170 DATA 10,F7,60,8A,48,AD,05,90,C9,DE,D0       <218>
175 DATA 03,20,18,E5,68,4C,3B,C4,20,73,00       <143>
180 DATA 08,C9,40,F0,04,28,4C,E7,C7,28,20       <180>
185 DATA 3C,15,4C,AE,C7,20,73,00,C9,91,D0       <205>
190 DATA 55,20,73,00,A2,04,BD,FB,17,9D,01       <186>
195 DATA 90,CA,D0,F7,AD,86,02,9D,00,96,E8       <250>
200 DATA D0,FA,A9,7F,AA,9D,00,16,E9,08,E9       <031>
205 DATA 00,29,7F,CA,10,F4,60,20,9E,D7,8A       <233>
210 DATA 10,F9,4C,48,D2,A2,00,86,FD,20,6A       <234>
215 DATA 15,48,4A,4A,4A,4A,66,FD,09,18,85       <237>
220 DATA FE,20,FD,CE,20,6A,15,A8,68,29,07       <014>
225 DATA AA,A9,00,38,6A,CA,10,FC,60,C9,9C       <038>
230 DATA D0,19,20,73,00,A2,18,86,23,A0,00       <171>
235 DATA 84,22,98,91,22,C8,D0,FB,E6,23,E8       <009>
240 DATA E0,20,90,F4,60,C9,8E,D0,06,20,73       <234>
245 DATA 00,4C,18,E5,20,6A,15,08,20,FD,CE       <005>
250 DATA 20,73,15,28,F0,04,11,FD,D0,04,49       <220>
255 DATA FF,31,FD,91,FD,60,20,73,00,20,FA       <035>
260 DATA CE,20,73,15,31,FD,F0,02,A9,01,20       <253>
265 DATA 94,D7,20,B4,DF,4C,F7,CE,A9,00,85       <082>
270 DATA 0D,20,73,00,C9,40,F0,DD,20,79,00       <250>
275 DATA 4C,8D,CE,144,17,0,222                  <225>
```


Listing 1. »Mini-Grafik VC 20« (Basic-Lader)



Die »Mini-Grafik« erzeugt ein kleines Grafik-Fenster

```
10 @ON:@CLR:DIM P(127)                          <236>
20 FOR I=0 TO 127                               <254>
30 @1,0,I                                       <096>
40 NEXT                                         <170>
50 FOR I=1 TO 127                               <029>
60 @1,I,64                                      <184>
70 NEXT                                         <200>
80 FOR X=0 TO 127                               <073>
90 @1,X,(1-SIN(X*π/64))*63.5                    <074>
95 @1,X,(1-COS(X*π/64))*63.5                    <078>
100 NEXT                                        <230>
180 FOR X=0 TO 127                              <173>
190 @1,X,(SIN(X*π/16))*(127-X)/2+63.5           <070>
200 NEXT                                        <074>
250 FOR I=1 TO 127                              <229>
260 @1,I,0                                      <071>
265 @0,0,I                                      <075>
270 NEXT                                        <145>
280 REM                                         <168>
281 Y=RND(TI)↑2*128                             <245>
285 P(Y)=P(Y)+1 :IF P(Y)>127 THEN 400           <228>
286 @1,Y,P(Y)                                   <059>
300 GOTO 280                                    <080>
400 FOR I=6144 TO 8191                          <083>
420 POKE I,255-PEEK(I):NEXT                     <016>
900 @RETURN                                     <085>
```


Listing 2. Demo-Programm zur »Mini-Grafik«

6510 — Die Suche nach dem Prozessor

Mit einem Raumschiff fliegen Sie durch ein Höhlenlabyrinth. Ihre Aufgabe: Finden Sie den Prozessor 6510 und setzen ihn in seinen Sockel. Wem das Höhlen-System nicht gefällt, der kann sich selbst neue erstellen.

Ziel des Spiels ist es, einen IC am Ende eines Labyrinthes aufzunehmen, und diesen in seine Fassung einzusetzen. Es gehört viel Geschicklichkeit dazu, die sechs Labyrinthe zu durchqueren. Außerdem machen einem Hindernisse wie Laser, Gravitation und Fuel das Leben schwer. Mit den Funktionstasten kann man die Geschwindigkeit, die Stärke der Gravitation, Anzahl der Schiffe, sowie die Fuelabnahme einstellen. Zu Beginn befindet man sich in einem kuppelähnlichem Raum, in dem sich die IC-Fassung befindet. Durch Drücken des Feuerknopfes kann das Fahrwerk ein- beziehungsweise ausgefahren werden. Unfallfreies Landen ist jedoch nur auf den dafür vorgesehenen Fuel-Plattformen zum Auftanken möglich.

Zum Eintippen:

Da das Lesen der Daten mit Prüfsummenroutine viel mehr Zeit beansprucht, als ohne diese, empfiehlt es sich, sämtliche »GOSUB3440« nach dem ersten einwandfreien Durchlauf zu entfernen. Ebenso ist es notwendig, alle Prüfsummen, das heißt DATAs über 255 zu entfernen. Durch das Entfernen des REMs aus Zeile 810, entfällt das blinde Eintippen von »POKE648,4« nach dem Drücken von »RUN/STOP + RESTORE«. Dieser »POKE« wäre notwendig, um wieder auf den normalen Bildschirm zu kommen. Falls jemandem die Labyrinthe zu schwierig sein sollten, können diese geändert werden.

Eine Besonderheit an »6510« ist unter anderem die Unterbringung der Fuelsäule. Diese befindet sich außerhalb des Bildschirmfensters. Dieses wurde mit Hilfe des IRQ (Interrupt Request), des Rasterzeileninterrupts verwirklicht. Mit einer ähnlichen Interruptroutine wurde auch die Steuerung und der Ablauf der Melodie möglich.

Das Prinzip eines solchen Interrupts ist folgendes:

Das Bild, das man auf einem Fernsehgerät oder einem Monitor sieht, wird aus vielen einzelnen Zeilen zusammengesetzt (Rasterzeilen). Diese Zeilen werden beim Fernseher zirka 25 mal pro Sekunde aufgebaut, so daß ein flimmerfreies Bild entsteht. Beim C 64 übernimmt der VIC (Video-Interface-Chip) diese Aufgabe. Wird nun eine ganz bestimmte Zeile aufgebaut, verzweigt der Prozessor zu einer Interruptroutine. Bei »6510« wird in dieser Routine die Rahmenfarbe ab einer bestimmten Rasterzeile auf den Wert 5 (grün) beziehungsweise den Wert 0 (schwarz) gesetzt. Die Rasterzeile, ab der auf Grün geschaltet wird, ist von der Menge des vorhandenen Fuels abhängig. Mit dem Timer-Interrupt wird die Hauptschleife gesteuert. Dieser Interrupt wird unter anderem durch den in der Speicherstelle 56549 abgelegten Wert geregelt. Hierdurch besteht auch die Möglichkeit, die Geschwindigkeit des Cursors zu manipulieren (TI$ wird ebenfalls beeinflußt). Der Timer-Interrupt wird von den CIAs (Complex Interface Adapter) ausgelöst. Diese zählen von dem in 56549 stehendem Wert auf Null herunter, wonach sie einen Interrupt auslösen und wieder mit dem Zählen beginnen.

Die Daten für die Melodie werden ab $C500 bis $C6FF abgelegt.

Hinweise zum Ändern und Hinzufügen von Labyrinthen

Um überhaupt Labyrinthe ändern oder hinzufügen zu können, muß vor die POKEs in Zeile 810 ein »REM« gesetzt werden. Danach startet man das Programm. Sobald das Auswahlmenü erscheint, drückt man die RUN/STOP-Taste. Nun hat man den umdefinierten Zeichensatz zur Verfügung.

Ändern von Labyrinthen

Die Labyrinthe liegen in den Zeilen 3530 bis 5170. Sollen Labyrinthe geändert werden, so listet man das zu ändernde Labyrinth (siehe Listing). Nun kann das Labyrinth wie ein normales Basic-Programm editiert werden.

Hinzufügen von Labyrinthen

Das Hinzufügen von Labyrinthen ist etwas komplizierter. Zuerst sind im Programm folgende Änderungen nötig (angegeben ist jeweils die gesamte Zeile):

1) Zeile 400:

400 RI=0:CO=0:CK=1:V=53248:SI=54272:LM=6+Anzahl der hinzugefügten Labyrinthe

2) Zeile 1810:

1810 ON LA GOSUB 3520, 3750, Zeile1, Zeile2, ..., Zeile n,4020,4280,4550,4890

Dabei entspricht »Zeile1« der Anfangszeile des ersten hinzugefügten Labyrinths, »Zeile2« der Anfangszeile des 2. hinzugefügten Labyrinths und so weiter.

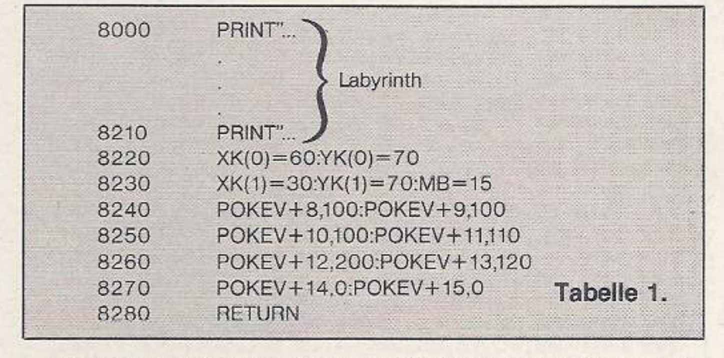
3) Zeile 2060:

2060 A=0:ON LA GOTO 2190,2070,...,2070,2070,2700, 2080

Nach der ersten »2070« müssen soviele »2070« eingefügt werden, wie neue Labyrinthe hinzugefügt wurden. Nicht vergessen, die »2070« durch Kommata zu trennen!

Jetzt können die neuen Labyrinthe an den Schluß des Programms (ab Zeile 8000) angefügt werden. Dabei ist folgendes Format einzuhalten:

22 PRINT-Zeilen, in denen das Labyrinth gePRINTet wird. In den nachfolgenden Zeilen müssen die Koordinaten des Raumschiffs, der Laser und der Fuel-Plattform gesetzt werden (Tabelle 1). Die X-Koordinaten dürfen den Wert 255 nicht überschreiten. Dies demonstrieren wir am besten an einem Beispiel.



```
8000   PRINT"...  }
       .          } Labyrinth
       .          }
8210   PRINT"...  }
8220   XK(0)=60:YK(0)=70
8230   XK(1)=30:YK(1)=70:MB=15
8240   POKEV+8,100:POKEV+9,100
8250   POKEV+10,100:POKEV+11,110
8260   POKEV+12,200:POKEV+13,120
8270   POKEV+14,0:POKEV+15,0
8280   RETURN
```

Tabelle 1.

Die Bedeutung der Variablen und POKEs ist in Tabelle 1 erläutert. Der Variablen MB muß immer der Wert 15 zugeordnet werden.

Soll keine Fuel-Plattform erscheinen, so sind die Koordinaten auf 0 zu POKEn. Außerdem ist darauf zu achten, daß sich keine Sprites überschneiden, da sonst die Kollisionsabfrage nicht einwandfrei arbeitet.

Hinweise zum Erstellen eines Labyrinths

Erst wird — wie bereits beschrieben — der Zeichensatz eingeschaltet. Dann löscht man den Bildschirm und baut das Labyrinth aus den umdefinierten Zeichen auf. Dabei ist zu beachten, daß der Bildschirm nicht nach unten scrollt (dies kann pas-

sieren, wenn man ein Zeichen an die letzte Position der Zeile setzt). Dieses Scrollen kann verhindert werden, indem man das Zeichen an die vorletzte Position der Zeile setzt, den Cursor vor das Zeichen positioniert und mit der INSERT-Funktion um eine Stelle nach rechts verschiebt. Außerdem müssen die untersten drei Zeilen frei bleiben (SCORE, TIME...). Auch darf das Raumschiff weder den oberen noch den unteren Bildschirmrand verlassen, da dies nicht in der Hauptschleife überprüft wird. Ist das Labyrinth fertig erstellt, muß es in das Programm eingefügt werden. Dazu wird der Cursor in die linke obere Ecke des Bildschirms gebracht. Dann wird mit »INSERT« Platz für die Zeilennummer, ein Fragezeichen für »PRINT« und ein Anführungszeichen geschaffen. Nun wird die Zeilennummer, das Fragezeichen und das Anführungszeichen in den gerade geschaffenen Platz eingefügt. Nun drückt man »RETURN«. Jetzt fährt man mit dem Cursor vorsichtig an den unteren Rand des Bildschirms und läßt die obere Zeile aus dem Bildschirmfenster nach oben herausscrollen. Mit den verbliebenen Zeilen verfährt man genauso. Man sollte allerdings nicht vergessen, die Zeilennummer laufend zu erhöhen.

(Harald Beine/Arne Jansen/rg)

VIC Register	Bedeutung
0	X-Koordinate des Raumschiffs Sprite 1
1	Y-Koordinate des Raumschiffs Sprite 1
2	X-Koordinate des Raumschiffs Sprite 2
3	Y-Koordinate des Raumschiffs Sprite 2
4	X-Koordinate des Fahrwerks
5	Y-Koordinate des Fahrwerks
6	frei
7	frei
8	X-Koordinate des 1. Lasers
9	Y-Koordinate des 1. Lasers
10	X-Koordinate des 2. Lasers
11	Y-Koordinate des 2. Lasers
12	X-Koordinate der Fuel-Platform
13	Y-Koordinate der Fuel-Platform
14	X-Koordinate des Zusatzsprites
15	Y-Koordinate des Zusatzsprites

Zusatzsprite = IC-Fassung, Schlüssel, 6510

Bit-Belegung des VIC

Variable	Bedeutung
XK(0)	X-Koordinate des Raumschiffs auf der Hinfahrt
YK(0)	Y-Koordinate des Raumschiffs auf der Hinfahrt
XK(1)	X-Koordinate des Raumschiffs auf der Rückfahrt
YK(1)	Y-Koordinate des Raumschiffs auf der Rückfahrt
MB	MSB des Raumschiffs auf der Rückfahrt (0 oder 15)

Maschinenroutinen: Aufruf mit SYS

704	Ersetzen von »POKE648,4«
832	während des Datenlesens: Kopieren des Zeichensatzes ins RAM
	im Hauptprogramm: Schlußszene
32768	Initialisieren der Hauptschleife, wählen des Raster-IRQ
	Hauptschleife: Joystickabfrage
	Bewegung des Raumschiffs
	Gravitation
	Fuel
	Laser
	Musik
33217	Joystickabfrage ausschalten
33204	Joystickabfrage einschalten
52736	Hauptschleife ausschalten

Maschinenroutinen von »6510«

Die wichtigsten Variablen:

LS	Lasergeschwindigkeit
SH	Anzahl Schiffe
MU	Musik ein beziehungsweise aus
FS	Fuelsäulengeschwindigkeit
LE	Level
RI	Richtung
V	VIC-Basisadresse
SI	SID-Basisadresse
X	Speicherstelle X-Koordinate
Y	Speicherstelle Y-Koordinate
LA	Aktuelles Labyrinth

Variablenliste von »6510«

Unterroutinen:

100— 820	Einleseroutine
830—1550	Copyright und Titelbild
1600—1760	Initialisieren des Speichers
1770—2050	Hauptprogramm
2060—2070	Kollision des Raumschiffs mit Sprite 7
	Sprite 7:
	Chipfassung, Schlüssel, Chip
2080—2180	Chip genommen
2190—2690	»Durchgekommen«
2700—2820	Schlüssel genommen
2830—2930	Rüttelroutine
2940—3070	Game Over
3120—3130	Cursor Positionieren
3140—3150	Joystickabfrage aus
3160—3170	Joystickabfrage ein
3190—3360	VIC und SID initialisieren
3410—3510	Aktuelle DATA-Zeile und Prüfsummenroutine
3530—5170	Labyrinthe 1 bis 6
5180—Ende	DATAs

Unterroutinen von »6510«

```
100 REM  ******************************          <223>
110 REM  *            6510              *          <029>
120 REM  ******************************          <243>
130 REM  *                              *          <101>
140 REM  * HARALD BEINE & ARNE JANSEN *          <137>
150 REM  *  SCHOETTELKOTTER  DAMM 13  *          <128>
160 REM  *         4432   GRONAU       *          <028>
170 REM  *                              *          <141>
180 REM  ******************************          <047>
190 REM                                           <077>
200 REM  ******************************          <067>
210 REM  *       STEUERZEICHEN         *          <147>
220 REM  *  ----------------------     *          <247>
230 REM  *  "{CLR}"  =  SHIFT + CLR/HOME  *       <191>
240 REM  *  "{HOME}"  =  CLR/HOME        *        <160>
250 REM  *  "{DOWN}"  =  CRSR DOWN       *        <225>
260 REM  *  "{WHITE}"  =  CTRL  + 2       *       <000>
270 REM  *  "{RED}"  =  CTRL  + 3       *         <034>
280 REM  *  "{CYAN}"  =  CTRL  + 4       *        <176>
290 REM  *  "{PURPLE}"  =  CTRL  + 5       *      <184>
300 REM  *  "{GREEN}"  =  CTRL  + 6       *       <069>
310 REM  *  "{BLUE}"  =  CTRL  + 7       *        <081>
320 REM  *  "{YELLOW}"  =  CTRL  + 8       *      <219>
330 REM  *  "{RVSON}"  =  CTRL  + 9       *       <090>
340 REM  *  "{ORANGE}"  =  C=    + 1       *      <022>
350 REM  *  "{LIG.BLUE}"  =  C=    + 7       *    <063>
360 REM  ******************************          <228>
370 REM                                           <002>
380 POKE 56,126:CLR:POKE 251,0                    <223>
390 R=1:LS=1:SH=3:MU=1:SP=100:MS= 5:FS=1:OF=0:LE=1  <231>
400 RI=0:CO=0:CK=1:V=53248:SI=54272               <059>
410 POKE SI+12,71:POKE SI+13,25:POKE SI+11,0      <082>
420 FOR I=0 TO 17:READ A:GOSUB 3440:POKE 52736+I,A:N
    EXT                                           <229>
430 SYS 52736:POKE SI+24,0:POKE V+21,0            <189>
440 POKE V+48,0:POKE V+17,155:POKE V+16,0         <172>
450 POKE V+32,0:POKE V+33,0                       <174>
460 CX= 0:CY=10:GOSUB 3120                        <192>
470 PRINT CHR$(14)CHR$(8)                         <061>
480 PRINT"{CLR,GREEN,RVSON,40SPACE}";             <188>
490 PRINT" BITTE 154 SEK. GEDULD !!{2SPACE}LESE DATE
    N !!";                                        <116>
500 PRINT"{40SPACE}"                              <210>
510 X=248:Y=249                                   <139>
520 REM DATEN EINLESEN                            <088>
530 FOR I=832 TO 832+33:READ A:GOSUB 3440:POKE I,A:N
    EXT                                           <085>
540 SYS 832                                       <089>
550 FOR T=0 TO 62:READ A:GOSUB 3440:POKE 61440+T,A:N
    EXT                                           <118>
560 FOR T=0 TO 62:READ A:GOSUB 3440:POKE 61504+T,A:N
    EXT                                           <129>
570 FOR T=0 TO 62:READ A:GOSUB 3440:POKE 61568+T,A:N
    EXT                                           <149>
580 FOR T=0 TO 62:POKE 61696+T,0:NEXT             <252>
590 FOR T=0 TO 62 STEP 3:POKE 61696+T,60:NEXT     <024>
600 FOR T=0 TO 62:READ A:GOSUB 3440:POKE 61760+T,A:N
    EXT                                           <173>
610 FOR T=0 TO 62:READ A:GOSUB 3440:POKE 61824+T,A:N
    EXT                                           <184>
620 FOR T=0 TO 62:READ A:GOSUB 3440:POKE 61888+T,A:N
    EXT                                           <204>
```

Listing »6510«. Beachten Sie den Checksummer 64

```
630 FOR T=0 TO 62:READ A:GOSUB 3440:POKE 61952+T,A:NEXT <206>
640 FOR I=0 TO 25:READ A:GOSUB 3440:B$=B$+CHR$(A):NEXT <068>
650 FOR AN=0 TO 18:FOR I=0 TO 7 <076>
660 READ A:GOSUB 3440:POKE 59392+I+8*AN,A <074>
670 NEXT I:NEXT AN <182>
680 POKE 56334,PEEK(56334)AND 254:POKE 1,PEEK(1)AND 251 <141>
690 FOR A=0 TO 7:POKE 59552+A,PEEK(54088+A):NEXT <232>
700 FOR A=0 TO 7:POKE 59560+A,PEEK(55112+A):NEXT <230>
710 FOR A=0 TO 7:POKE 59568+A,PEEK(54008+A):NEXT <251>
720 FOR A=0 TO 7:POKE 59544+A,PEEK(55032+A):NEXT <253>
730 POKE 1,PEEK(1)OR 4:POKE 56334,PEEK(56334)OR 1 <243>
740 FOR I=32768 TO 33229:READ A:GOSUB 3440:POKE I,A:NEXT <234>
750 FOR I=36600 TO 36863:POKE I,0:NEXT <075>
760 FOR I=0 TO 255:READ A:GOSUB 3440:POKE 50432+I,A:READ A <171>
770 GOSUB 3440:POKE 50688+I,A:NEXT <085>
780 FOR I=0 TO 117:READ A:GOSUB 3440:POKE 828+I,A:NEXT: <085>
790 FOR I=0 TO 15:READ A:GOSUB 3440:POKE 704+I,A:NEXT <235>
800 IF F=1 THEN END:REM WENN FEHLER IN DATA,DANN ENDE <175>
810 REM POKE770,704AND255:POKE771,704/256 <090>
820 POKE V+24,11:POKE 56576,PEEK(56576)AND 252:POKE 648,192 <111>
830 GOSUB 3190:PRINT"{CLR,LIG.BLUE}":POKE SI+24,0 <184>
840 PRINT"LPLLPLLOPQLLLLOLQOPQOLQOLOPLLOPQOPOLLOPL"; <154>
850 PRINT"ABFEABEABFEFABABFEABFEABEFFEABFFABFEABFE"; <250>
860 PRINT"CD{2SPACE}CD{6SPACE}CD{4SPACE}CD{2SPACE}CD{4SPACE}CD{6SPACE}CD{2SPACE}"; <040>
870 PRINT:PRINT <213>
880 PRINT"GH{6SPACE}GH{2SPACE}GH{2SPACE}GH{4SPACE}GH{4SPACE}GH{4SPACE}GH{4SPACE}"; <116>
890 PRINT"IJ@IJ K IJGHIJNJIJ{4SPACE}IJ@IJ@IJKIJ@IJKIJK"; <057>
900 PRINT"LPLLPLLOPQLLLLOLQB{4SPACE}AOLOPLLOPQOPOLLOPL"; <125>
910 IF CO=1 THEN 990 <218>
920 C=0:POKE V+21,3:POKE V+1,80:POKE V+3,80:POKE V+5,80 <027>
930 PRINT"{DOWN,YELLOW}":FOR I=1 TO 3:READ AW,EW,SW,S1,S2:GOSUB 1560 <218>
940 NEXT:FOR I=104 TO 245 <187>
950 POKE V+0,I+6:POKE V+2,I+6 <045>
960 IF I>=104 AND I=(INT(I/8))*8 AND C<=13 THEN GOSUB 3370 <094>
970 NEXT:PRINT:GOSUB 3390 <184>
980 FOR I=1 TO 2:READ AW,EW,SW,S1,S2:GOSUB 1560:NEXT <001>
990 IF CO=0 THEN 1030 <075>
1000 POKE V+0,172:POKE V+1,80:POKE V+2,172:POKE V+3,80 <111>
1010 PRINT"{2DOWN}"TAB(14)"{YELLOW}"A$ <108>
1020 GOSUB 3390 <091>
1030 CX=0: CY=16:GOSUB 3120 <003>
1040 REM     F        GRAVITATION        NORMAL <250>
1050 PRINT"{GREY 2,DOWN,SPACE}F1{4SPACE}: GRAVITATION{4SPACE}:{2SPACE}NORMAL{2SPACE}" <160>
1060 REM     F        LASER SPEED        NORMAL <174>
1070 PRINT" F3{4SPACE}: LASER{SHIFT-SPACE}SPEED{4SPACE}:{2SPACE}NORMAL{2SPACE}" <205>
1080 REM     F        ANZAHL SCHIFFE <199>
1090 PRINT" F5{4SPACE}: ANZAHL{SHIFT-SPACE}SCHIFFE :{2SPACE}3{2SPACE}" <155>
1100 REM     F        MUSIK              EIN <138>
1110 PRINT" F7{4SPACE}: MUSIK{10SPACE}:{2SPACE}EIN" <013>
1120 REM     CRSR ←  FUELSAEULE          SCHNELL <128>
1130 PRINT" CRSR ←: FUELSEULE{6SPACE}:{2SPACE}SCHNELL" <011>
1140 REM     RETURN  START <117>
1150 PRINT" RETURN: START " <135>
1160 FOR T=0 TO 30:NEXT:A=PEEK(203) <219>
1170 ON A GOSUB 1430,1240,1200,1270,1340,1400 <122>
1180 IF A=1 THEN POKE SI+24,15:GOTO 1600 <085>
1190 GOTO 1160 <251>
1200 CX=27:CY=20:GOSUB 3120 <225>
1210 IF MU=1 THEN PRINT"AUS":MU=0:RETURN:REM AUS <243>
1220 IF MU=0 THEN PRINT"EIN":MU=1:REM EIN <027>
1230 RETURN <096>
1240 CX=27:CY=21:GOSUB 3120 <010>
1250 IF FS=1 THEN PRINT"LANGSAM":FS=0:RETURN:REM LANGSAM <061>
1260 IF FS=0 THEN PRINT"SCHNELL":FS=1:RETURN:REM SCHNELL <083>
1270 GR=GR+1:IF GR>4 THEN GR=1 <031>
1280 CX=27:CY=17:GOSUB 3120 <056>
1290 IF GR=1 THEN PRINT"STARK{5SPACE}":REM STARK <237>
1300 IF GR=2 THEN PRINT"SCHWACH{2SHIFT-SPACE,SPACE}":REM SCHWACH <048>
1310 IF GR=3 THEN PRINT"NORMAL{4SPACE}":REM NORMAL <011>
1320 IF GR=4 THEN PRINT"SCHWAECHER":REM SCHWAECHER <062>
1330 RETURN <197>
1340 LS=LS+1:IF LS>3 THEN LS=1 <125>
1350 CX=27:CY=18:GOSUB 3120 <127>
1360 IF LS=1 THEN PRINT"SCHNELL ":REM SCHNELL <065>
1370 IF LS=2 THEN PRINT"LANGSAM ":REM LANGSAM <064>
1380 IF LS=3 THEN PRINT"NORMAL{2SPACE}":REM NORMAL <087>
1390 RETURN <001>
1400 SH=SH+1:IF SH>15 THEN SH=3 <222>
1410 CX=26:CY=19:GOSUB 3120 <187>
1420 PRINT SH CHR$(157)CHR$(32):RETURN <191>
1430 PRINT"{HOME,8DOWN}" <019>
1440 FOR T=0 TO 4 <052>
1450 PRINT"{37SPACE}" <140>
1460 NEXT <059>
1470 REM              JOYSTICK IN PORT 1 <207>
1480 PRINT"{11SPACE}JOYSTICK{SHIFT-SPACE}IN{SHIFT-SPACE}PORT{SHIFT-SPACE}1{7SPACE}" <007>
1490 FOR T=0 TO 7 <105>
1500 PRINT"{37SPACE}" <190>
1510 NEXT <109>
1520 AW=173:EW=255:SW=1:S1=0:S2=2:GOSUB 1560 <155>
1530 POKE V+16,3:AW=0:EW=100:GOSUB 1560 <188>
1540 POKE V+0,0:POKE V+2,0:POKE V+16,0 <032>
1550 RETURN <162>
1560 REM SCHLEIFE <240>
1570 FOR K=AW TO EW STEP SW:POKE V+S1,K:POKE V+S2,K <106>
1580 FOR T=0 TO 30:NEXT:NEXT <104>
1590 RETURN <202>
1600 IF LS=1 THEN POKE 33004,40:POKE 33020,20 <114>
1610 IF LS=2 THEN POKE 33004,200:POKE 33020,100 <218>
1620 IF LS=3 THEN POKE 33004,100:POKE 33020,50 <184>
1630 IF GR=1 THEN GR=30 <192>
1640 IF GR=2 THEN GR=150 <254>
1650 IF GR=3 THEN GR=70 <218>
1660 IF GR=4 THEN GR=255 <026>
1670 POKE 33032,230:POKE 33033,249 <119>
1680 POKE 36611,  1:POKE 36607,8 <198>
1690 POKE 33156,MS :POKE 36860,5 <062>
1700 IF FS=1 THEN POKE 32978,10 <137>
1710 IF FS=0 THEN POKE 32978,50 <150>
1720 POKE SI+2,2:POKE SI+3,7 <216>
1730 IF MU=0 THEN POKE 33152, 96:POKE SI+24,15 <189>
1740 POKE V+21,247 <149>
1750 POKE 56549,SP:POKE 36862,51 <187>
1760 SYS 32768:POKE V+30,0:POKE V+31,0 <175>
1770 IF RI=1 THEN L1=5:L2=1:L3=-1 <230>
1780 IF RI=0 THEN L1=1:L2=6:L3=1 <069>
1790 FOR LA=L1 TO L2 STEP L3:TI$="000000" <156>
1800 AX=PEEK(V+21):POKE V+21,0 <096>
1810 ON LA GOSUB 3520,3750,4020,4280,4550,4890 <112>
1820 POKE V+21,AX <226>
1830 POKE X,XK(RI):POKE Y,YK(RI) <189>
1840 POKE 36614,0:IF RI=1 THEN POKE 36614,MB <110>
1850 CX=0 :CY=22:GOSUB 3120 <055>
1860 PRINT"{WHITE,SPACE}{**********R************R***************S" <077>
1870 REM      SHIPS      TIME         SCORE <022>
1880 PRINT" -SHIPS:{3SPACE}-TIME:{6SPACE}-SCORE:{8SPACE}-" <110>
1890 PRINT" Z**********F************F***************X{HOME}" <165>
1900 CX=8 :CY=23:GOSUB 3120:PRINT SH;:GOSUB 3150 <171>
1910 GOSUB 3170:CX=31:GOSUB 3120:PRINT SC;:POKE 33028,GR <135>
1920 POKE 251,0:POKE V+30,0:POKE V+31,0 <051>
1930 SYS 8*4096 <230>
1940 IF PEEK(251)THEN 2780 <073>
1950 POKE 250,RI <154>
1960 A=PEEK(V+31):IF(A AND 7)THEN 2830 <202>
1970 A=PEEK(V+30) <034>
1980 IF A=68 THEN GOSUB 2940 <178>
1990 IF A AND 128 THEN 2060 <082>
2000 IF(A AND 7)THEN 2830 <078>
2010 IF PEEK(36862)>=250 THEN 2050 <240>
2020 IF(PEEK(56321)AND 16)=0 THEN GOSUB 3080 <210>
2030 CX=18:CY=23:GOSUB 3120:PRINT MID$(TI$,3,2); <204>
2040 PRINT":"RIGHT$(TI$,2):GOTO 1940 <224>
2050 POKE 53274,0:POKE 53280,0:POKE 33028,1:OF=1:GOTO 1940 <162>
2060 A=0:ON LA GOTO 2190,2070,2070,2070,2700,2080 <174>
2070 GOTO 2000 <105>
2080 RI=1:POKE V+14,0:POKE V+15,0:POKE 36610,1:POKE 33038,40 <059>
2090 A=0:IF U=1 THEN 2000 <187>
2100 GOSUB 3150 <146>
2110 POKE 648,4:FOR H=0 TO 5 :FOR I=7 TO 18 <151>
2120 CX=15:CY=I:GOSUB 3120 <097>
2130 PRINT"{RVSON,BLUE,23SPACE}" <104>
2140 CX=15:CY=I:GOSUB 3120 <117>
2150 PRINT"{RVSON,RED,23SPACE}" <121>
2160 CX=15:CY=I:GOSUB 3120 <137>
2170 PRINT"{RVSON,YELLOW,23SPACE}" <015>
2180 NEXT:NEXT:U=1:POKE 648,192:GOSUB 3170:GOTO 2000 <142>
2190 IF PEEK(36610)<>1 THEN 2000 <050>
2200 POKE V+23,176:POKE V+29,128:POKE 50174,200 <249>
2210 POKE V+12,176:POKE V+13,90 <056>
2220 POKE 33038,45:POKE V+21,195:POKE SI+24,0 <185>
2230 SYS 52736:POKE 32809,162:POKE 32810,128 <149>
2240 SYS 32768:POKE V+40,2:POKE V+32,0 <149>
2250 POKE V+26,0:FOR I=0 TO 62:POKE 62144+I,0:NEXT <252>
2260 FOR I=0 TO 3000:NEXT:GOSUB 3150 <059>
2270 FOR I=0 TO 2:CX=0:CY=22+I:GOSUB 3120 <139>
2280 PRINT"{39SPACE}"; <008>
2290 PRINT CHR$(20)" ";:NEXT <072>
2300 SYS 828:POKE V+45,1:POKE V+46,8 <122>
2310 POKE 50172,203:POKE 50173,203:POKE V+43,6 <146>
2320 POKE V+44,6:POKE V+8,152:POKE V+9,194 <027>
2330 POKE V+11,194:POKE V+10,216:POKE V+21,243:POKE 646,6 <169>
2340 FOR I=0 TO 62 STEP 3:POKE 62144+I,255 <078>
2350 FOR J=0 TO 50:NEXT J,I <104>
2360 FOR I=0 TO 4:CX=16:CY=18+I:GOSUB 3120 <036>
2370 PRINT"Q{7SPACE}Q";:NEXT <193>
2380 FOR I=194 TO 220 STEP 2:POKE V+9,I:POKE V+11,I <169>
2390 FOR J=0 TO 50:NEXT J,I <144>
2400 FOR I=0 TO 1:CX=16:CY=23+I:GOSUB 3120 <069>
```

```
2410 PRINT"Q{7SPACE}Q";:NEXT                          <233>
2420 POKE V+21,195:POKE V+23,128:POKE V+29,176        <069>
2430 POKE 50173,195:POKE 50172,195                    <132>
2440 POKE V+8,141:POKE V+10,189:POKE V+9,186          <247>
2450 POKE V+11,186:POKE V+21,243                      <088>
2460 CX=17:CY=17:GOSUB 3120:PRINT"{7SPACE}";          <041>
2470 FOR I=0 TO 29:POKE V+8,141-I:POKE V+10,189+I     <028>
2480 FOR J=0 TO 20:NEXT J,I                           <231>
2490 FOR I=0 TO 3000:NEXT                             <146>
2500 FOR I=PEEK(V)TO 176 STEP SGN(176-PEEK(V))        <084>
2510 POKE V,I:POKE V+2,I                              <177>
2520 FOR J=0 TO 20:NEXT J,I:FOR I=0 TO 1000:NEXT      <022>
2530 FOR I=PEEK(V+1)TO 246:POKE V+1,I:POKE V+3,I      <219>
2540 FOR J=0 TO 20:NEXT J,I                           <035>
2550 SP=SP-10:LE=LE+1:IF SP<=35 THEN SP=SP+10         <203>
2560 FOR I=0 TO 3000:NEXT:POKE V+21,0:PRINT CHR$(142
     )                                                <235>
2570 PRINT"{CLR,WHITE,4DOWN,7SPACE}HERZLICHEN GLUECK
     WUNSCH !!                                        <108>
2580 PRINT"{3DOWN,11SPACE}SIE HABEN IHRE              <115>
2590 PRINT"{DOWN,9SPACE}MISSION ERFOLGREICH           <064>
2600 PRINT"{2DOWN,14SPACE}BEENDET !!                  <072>
2610 PRINT"{3DOWN,6SPACE}AB JETZT WIRD'S SCHNELLER ! <175>
2620 PRINT"{2DOWN,15SPACE}LEVEL :"LE                  <136>
2630 FOR I=0 TO 5000:NEXT                             <033>
2640 GOSUB 3190                                       <180>
2650 PRINT CHR$(14)                                   <122>
2660 POKE V+21,243                                    <045>
2670 POKE 33038,40:POKE 36610,0                       <205>
2680 RI=0:GOTO 1750                                   <143>
2690 GOTO 2000                                        <215>
2700 GOSUB 3150:POKE V+14,0:POKE V+15,0:POKE V+12,25
     5                                                <013>
2710 FOR Z=9 TO 13:CX=31:CY=Z:GOSUB 3120:PRINT" "     <223>
2720 POKE SI+18,33:POKE SI+18,32                      <147>
2730 FOR I=0 TO 300:NEXT:NEXT                         <015>
2740 CX=36:CY=6:GOSUB 3120:PRINT" "                   <214>
2750 FOR Z=7 TO 16:CX=35:CY=Z:GOSUB 3120:PRINT"{2SPA
     CE}":POKE SI+18,33                               <030>
2760 POKE SI+18,32:FOR I=0 TO 300:NEXT:NEXT           <062>
2770 CX=36:CY=Z:GOSUB 3120:PRINT"{2SPACE}":GOSUB 317
     0:GOTO 2010                                      <048>
2780 IF RI=1 AND LA=1 THEN POKE 251,0:GOTO 1940       <209>
2790 SC=SC+((240-VAL(TI$))*10)                        <063>
2800 CX=30:CY=23:GOSUB 3120:PRINT"SCORE{SHIFT-SPACE}
     : "SC;:POKE 251,0                                <167>
2810 REM SCORE                                        <015>
2820 NEXT:GOTO 1770                                   <035>
2830 REM RUETTEL ROUTINE                              <243>
2840 KR=PEEK(V+17):GOSUB 3150:POKE SI+11,129:POKE SI
     +8,50                                            <158>
2850 POKE SI+12,16* 1+5                               <188>
2860 POKE SI+13,16* 1+1                               <195>
2870 FOR I=1 TO 10:POKE V+17,23:POKE SI+11,129:FOR T
     =0 TO 50:NEXT                                    <180>
2880 POKE V+17,16:POKE SI+11,128:FOR T=0 TO 50:NEXT:
     NEXT                                             <143>
2890 POKE V+17,KR:POKE SI+11,0                        <242>
2900 POKE 36862,51                                    <145>
2910 POKE SI+12,71:POKE SI+13,25:SH=SH-1:TI$="000000
     "                                                <090>
2920 IF SH=0 THEN SH=3:GOTO 3010                      <041>
2930 GOTO 1830                                        <210>
2940 GOSUB 3150:Y1=PEEK(Y):POKE SI+5,255:FOR I=0 TO
     500:NEXT                                         <174>
2950 IF OF=1 THEN OF=0:POKE 53274,241:POKE 33028,GR:
     POKE 36862,250                                   <126>
2960 FOR I=PEEK(36862)TO 51 STEP-1:POKE 36862,I:POKE
      SI+ 8,I-51                                      <230>
2970 POKE SI+11,17:POKE SI+11,18:NEXT                 <066>
2980 POKE Y,Y1-1:GOSUB 3180:FOR XY=0 TO 100           <071>
2990 NEXT:IF PEEK(V+30)THEN Y1=Y1-1:GOTO 2980         <085>
3000 GOSUB 3170:POKE SI+11,0:A=0:POKE SI+5,77:RETURN <251>
3010 SYS 52736:POKE V+21,0:POKE 53280,0:POKE SI+4,0  <216>
3020 PRINT CHR$(147):CX=15:CY=12:GOSUB 3130:PRINT"GA
     ME OVER"                                         <247>
3030 IF SC>HS THEN HS=SC                              <216>
3040 PRINT"{3DOWN,9SPACE}SCORE : "SC:REM SCORE        <012>
3050 PRINT"{DOWN,4SPACE}HIGH SCORE : "HS:REM HIGH -  <234>
3060 FOR I=0 TO 4000:NEXT:RI=0:SC=0                   <056>
3070 POKE V+21,3:CO=1:SP=100:GOTO 830                 <143>
3080 IF((PEEK(V+21))AND 4)=4 THEN 3100                <235>
3090 POKE V+21,PEEK(V+21)OR 4:GOTO 3110               <038>
3100 POKE V+21,PEEK(V+21)AND 251                      <011>
3110 RETURN                                           <192>
3120 REM CURSOR POSITIONIEREN                         <145>
3130 POKE 211,CX:POKE 214,CY:SYS 58640:RETURN         <015>
3140 REM RAUMSCHIFF FESTHALTEN                        <181>
3150 SYS 33217:POKE V+21,PEEK(V+21)AND 207:RETURN     <222>
3160 REM RAUMSCHIFF LOSLASSEN                         <143>
3170 SYS 33204:POKE 36608,50:RETURN                   <057>
3180 POKE V+1,Y1:POKE V+3,Y1:POKE V+5,Y1:POKE V+7,Y1
     :RETURN                                          <242>
3190 REM SPRITES INITIALISIEREN                       <084>
3200 POKE V+31, 0:POKE 33048,4                        <022>
3210 POKE V+39, 7:POKE V+40 ,2                        <143>
3220 POKE V+41, 1:POKE V+42 ,0                        <140>
3230 POKE V+43, 5:POKE V+44 ,5                        <163>
3240 POKE V+32, 0:POKE V+33 ,0                        <159>
3250 POKE V+23,48:POKE V+27 ,240                      <078>
3260 POKE V+29,64:POKE 33052,84                       <198>
3270 FOR T=0 TO 62:POKE 61632+T,0:NEXT                <126>
3280 FOR T=0 TO 23:POKE 61632+T,255:NEXT              <241>
3290 POKE 50168,192:POKE 50169,193                    <232>
3300 POKE 50170,194:POKE 50171,195                    <232>
3310 POKE 50172,196:POKE 50173,196                    <249>
3320 POKE 50174,197:POKE 50175,198:POKE 36610,0       <055>
3330 POKE SI+ 2,  2:POKE SI+ 3,  7:U=0                <154>
3340 POKE SI+ 5, 77:POKE SI+19,0:POKE SI+20,244       <225>
3350 POKE SI+15, 30:POKE 36614,0:POKE V+16,0          <186>
3360 POKE SI+6,24:POKE SI+24,15:RETURN                <137>
3370 READ ZE:PRINT TAB(I/8)CHR$(ZE);:C=C+1:A$=A$+CHR
     $(ZE)                                            <128>
3380 RETURN                                           <207>
3390 PRINT"{ORANGE,DOWN,13RIGHT,SPACE}(C) 1984{2SPAC
     E}BY "                                           <184>
3400 PRINT"{DOWN,5RIGHT,2SPACE}"B$:RETURN             <002>
3410 REM AKTUELLE DATAZEILE                           <216>
3420 CZ=PEEK(63)+PEEK(64)*256:IF CK=1 THEN AZ=CZ:RET
     URN                                              <051>
3430 RETURN                                           <001>
3440 REM PRUEFSUMMENROUTINE                           <137>
3450 GOSUB 3410:CK=0                                  <134>
3460 IF A>255 THEN CS=A:GOTO 3480                     <108>
3470 S=S+A:RETURN                                     <166>
3480 IF S<>CS THEN GOSUB 3420:PRINT"FEHLER IN :"AZ"-
     "CZ;CS-S                                         <144>
3490 IF S<>CS THEN READ A:F=1:S=0:CK=1:RETURN         <059>
3500 IF S=CS THEN GOSUB 3420:PRINT"ZEILE :"AZ"-"CZ"O
     K"                                               <173>
3510 S=0:CK=1:READ A:RETURN                           <109>
3520 REM ** LABYRINTHE **                             <246>
3530 PRINT"{CLR,BLUE,2DOWN,18SPACE}UQQQS              <001>
3540 PRINT"{16SPACE}UQQT VQQS                         <050>
3550 PRINT"{15SPACE}UQT{5SPACE}VQS                    <154>
3560 PRINT"{14SPACE}UQT{7SPACE}VQS                    <164>
3570 PRINT"{14SPACE}QT{9SPACE}VQ                      <006>
3580 PRINT"{13SPACE}UQ{11SPACE}QS                     <014>
3590 PRINT"{13SPACE}QT{11SPACE}VQ                     <027>
3600 PRINT"{12SPACE}UQ{13SPACE}QS                     <035>
3610 PRINT"QQQQQQQQQQQQT{13SPACE}VQQQQQQQQQQQ";       <211>
3620 PRINT"ABABFAPLDFAB{WHITE}Q{BLUE,15SPACE,WHITE}Q
     {BLUE}FLOLPLFEABF";                              <135>
3630 PRINT"CDCD{2SPACE}AB{2SPACE}CD{18SPACE}FPLOB{2S
     PACE}CD ";                                       <102>
3640 PRINT"{6SPACE}CD{23SPACE}ABF{6SPACE}";           <174>
3650 PRINT"{40SPACE}";                                <104>
3660 PRINT"GH{13SPACE}GH{7SPACE}GH{10SPACE}GH{2SPACE
     }";                                              <174>
3670 PRINT"IJ@KIJ{2SPACE}KIJ@{WHITE}Q{BLUE,SPACE}@IJ
     {7SPACE}IJKK{WHITE}Q{BLUE,6SPACE}@IJ@ ";         <068>
3680 PRINT"QQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQ"
     ;                                                <046>
3690 POKE 36614,0:XK(0)=174:YK(0)=130                 <192>
3700 POKE V+8,248:POKE V+9,139:POKE V+12,0            <126>
3710 POKE V+13,0:POKE V+10,120:POKE V+11,139          <208>
3720 POKE V+23,176:POKE V+29,192                      <096>
3730 POKE 50175,198:POKE V+14,164:POKE V+15,73        <209>
3740 XK(1)=54:YK(1)=151:MB=15:POKE V+46,8:RETURN      <219>
3750 PRINT"{CLR,RED}OLLQLPOLQLOQPLLLLLLLOQLLPLLOLQLL
     PLLOQLLL";                                       <155>
3760 PRINT"LOLLPLLOQLLOQOLLLLQLLOLPLLLLOLQPLLLQOLLL"
     ;                                                <240>
3770 PRINT"OLOPLLQLOLLPLLLOLLQLLPOLLQLLQOPLLLLLPOLL"
     ;                                                <253>
3780 PRINT"LOQLLQLLLOPQLLLPLQLLPQLOLLQLLLOLLLLPLLOL"
     ;                                                <007>
3790 PRINT"LLOLLPOLLPOLLLPOLQLPLLQPOQLLQLPOLLPOLLLO"
     ;                                                <028>
3800 PRINT"LLLOPLLOPLLOLPLLOLPOLLPLLOLQLPLPOLLLOLLP"
     ;                                                <027>
3810 PRINT"LLLQLLQLLOQLLQLPLLQLLPLQLPOLLLPLLOLLLOLL"
     ;                                                <034>
3820 PRINT"EFABE{WHITE}Q{RED}ABPLLPLLPOLQLABEF{WHITE
     }Q{RED}ABFABFEABFEABFEA";                        <102>
3830 PRINT"{2SPACE}CD{2SPACE}CDFABOLLOLOBFCD{3SPACE}
     CD{5SPACE}CD{2SPACE}CD{2SPACE}C";                <171>
3840 PRINT"{9SPACE}CDABLPLO{8SPACE,SHIFT-SPACE}       <171>
3850 PRINT"{11SPACE}CDFOLE{21SPACE}";                 <222>
3860 PRINT"{2SPACE}IJK{11SPACE}AB{5SPACE}IJ{WHITE}Q{
     RED}J@IJ{5SPACE}IJ@GH@ ";                        <096>
3870 PRINT"LQOLL{WHITE}Q{RED,SPACE}@GH{SHIFT-SPACE,3
     SPACE}CD{4SPACE}KOLPOLLL{5SPACE}OLOPLQL";        <189>
3880 PRINT"LLLOLQLLIJ@{8SPACE}ILPLLLOPL{5SPACE}LLLOL
     LL";                                             <085>
3890 PRINT"LLLPOLQPLLL@{6SPACE}ILLOLLLLOL£££££LLPOL
     L";                                              <075>
3900 PRINT"LLPLQOPQLLLPL@ GH@POLLOLLLPLLLLLLLPLLOL"
     ;                                                <004>
3910 PRINT"LPOLLQOLLLPLLL@IJLQLOLPOQPLLLLOLPLPOLLLO"
     ;                                                <120>
3920 PRINT"PLLOLLLOLQOLLQPLLQLQPOLPLOLLLLPPLLLLOLLL"
     ;                                                <149>
3930 PRINT"OLQLOLLLOLLOLPLQLLQPLQOOLLOLLLPOLLLLPOLL"
     ;                                                <160>
3940 PRINT"LOLLPOLLLOLPOQOLLLPOLLPOLLLLLPLLOLLPLLOL"
     ;                                                <163>
3950 PRINT"LLOPLLOLQLOLLOQOLLLLPLLLLLLLLLLOLLLLLL"
     ;                                                <153>
3960 PRINT"LLPOLLLOLLLOLLLPOLLLPOLLLLLLLLLLLLLLLLL" <095>
3970 POKE 36614,0:XK(0)=26:YK(0)=116:XK(1)=59         <040>
3980 YK(1)=117:MB=15:POKE V+46,0                      <132>
3990 POKE V+8,64:POKE V+9,106:POKE V+10,208           <207>
4000 POKE V+11,106:POKE V+14,0:POKE V+15,0            <144>
4010 POKE V+12,245:POKE V+13,150:RETURN               <057>
4020 PRINT"{CLR,PURPLE}ABEFABAPOLPLOQLPLOLLPOPLLLPLQ
     LLOPLQLQOPL";                                    <238>
4030 PRINT"CD{2SPACE}CD OQPOLPOQOLLOPLLOLLPOLLQLLOQL
     QLQOL";                                          <008>
```

Listing »6510« (Fortsetzung)

```
4040 PRINT"{7SPACE}FOLPLPAB{SPACE,WHITE}Q{PURPLE}FLA
     BFLOLLLOLQLABABABABO";                         <158>
4050 PRINT"{7SPACE}GPPOLB{5SPACE}FCD FABNLPOLLCDCD{2
     SPACE}C L";                                    <193>
4060 PRINT"GH{5SPACE}IOOPOM{12SPACE}NPLLOM{9SPACE}"; <055>
4070 PRINT"IJ@{4SPACE}APPOB{13SPACE}FLLOM{10SPACE}"; <204>
4080 PRINT"OQLJ{4SPACE}FLPJ{7SPACE}GH{4SPACE}FLLOM{1
     0SPACE}";                                      <130>
4090 PRINT"LOPB{4SPACE}IPLB{4SPACE,WHITE}Q{PURPLE}IJ
     IJ{5SPACE}LLLM{10SPACE}";                      <190>
4100 PRINT"LPOJ{4SPACE}AOLJ{4SPACE}OQPLPM{4SPACE}FLM
     {11SPACE}";                                    <063>
4110 PRINT"PLQLJ{4SPACE}POB{4SPACE}LPLPOL{5SPACE}LM{
     11SPACE}";                                     <006>
4120 PRINT"ABFEF{4SPACE}OLD{3SPACE}NQLPLLM{5SPACE}NM <209>
4130 PRINT"CD{7SPACE}AB{4SPACE}NLPOBFE{4SPACE}NM{12S
     PACE}";                                        <245>
4140 PRINT"{SHIFT-SPACE,8SPACE}CD{3SPACE}NQLLF{7SPAC
     E}LM{12SPACE}";                                <145>
4150 PRINT"{13SPACE}GOLBF{7SPACE}NLMGH{10SPACE}";   <062>
4160 PRINT"{5SPACE}GH{6SPACE}@PLM{5SPACE}@IJ@OMIJ{9S
     PACE}L";                                       <174>
4170 PRINT"{5SPACE}IJ@GH @IPOM{4SPACE}IQLOLLLOLQLQQM
     {4SPACE}NQ";                                   <040>
4180 PRINT"QLPQOLLLIJ@OPLPM{4SPACE,WHITE}Q{PURPLE}BF
     EF AB E AB{5SPACE}AB";                         <027>
4190 PRINT"LPLLPLQPLQPLPLPM{15SPACE}CD{5SPACE}CD";  <119>
4200 PRINT"POLQLOPQLPQQOLQM{24SPACE}";              <127>
4210 PRINT"LLLQLLOQPLQLPOLLGH{6SPACE}GH{2SPACE}GH{5S
     PACE}GH GH";                                   <066>
4220 PRINT"LLLLQLLOLLLPLLOLIJ@{SPACE,WHITE}Q{PURPLE,
     2SPACE}@IJ@ IJ @IJ IJ@IJ";                     <023>
4230 PRINT"OLLLLLLPOLPLLLLOLLLLLLLLLLLLLLLLLLLLLLL"
     ;                                              <159>
4240 POKE 36614,0:XK(0)=30:YK(0)=65:POKE V+12,18    <079>
4250 POKE V+13,160:POKE V+14,0:POKE V+8,152         <199>
4260 POKE V+10,184:POKE V+11,175:POKE V+15,0        <004>
4270 XK(1)=57:YK(1)=102:MB=15:POKE V+9,70:RETURN    <236>
4280 PRINT"{CLR,LIG.BLUE}ABFBFEAOLBABOLQOPQOQLLQPLLQ
     QLPLQOBABF EF";                                <081>
4290 PRINT"CD{5SPACE}AB CDAOLPLQLOLQLOLQLLPBAB{2SPAC
     E}CD                                           <139>
4300 PRINT"{14SPACE}FQLLABOLQPLLQPL                 <010>
4310 PRINT"GH{2SPACE}GH{9SPACE}F{SPACE,WHITE}Q{LIG.B
     LUE}CDAOPOLQPLB{9SPACE}GH";                    <018>
4320 PRINT"IJ@ IJ@{3SPACE}@{10SPACE}PLQOPFF{5SPACE}@
     @ @ IJ";                                       <089>
4330 PRINT"QOLQPLLQLPLL{9SPACE}FLQQO@{5SPACE}NQPQOQL
     P";                                            <251>
4340 PRINT"LPLBABCD{3SPACE}A{WHITE}Q{LIG.BLUE,9SPACE
     }ABFLO@GH{3SPACE}PQQLOPQ";                     <226>
4350 PRINT"PLB CD{19SPACE}FADIJ{3SPACE}FAOQLOL";    <002>
4360 PRINT"OLJ{2SPACE,SHIFT-SPACE,20SPACE}NQLB{SHIFT
     -SPACE,4SPACE}QOQPO";                          <019>
4370 PRINT"LOLJ{4SPACE}@{8SPACE,WHITE}Q{LIG.BLUE,SPA
     CE}IJ{5SPACE}NLF{4SPACE}@ILPPQL";              <033>
4380 PRINT"QPOL{3SPACE}NPJGH{SHIFT-SPACE,3SPACE}NOLL
     PJ{4SPACE}LF{4SPACE}ILOPPOLQ";                 <105>
4390 PRINT"PFAB{4SPACE}FLPJ{5SPACE}NOPLB{3SPACE}@L@{
     4SPACE}ALPPOQOL";                              <067>
4400 PRINT"B CD{5SPACE}NLQ{WHITE}Q{LIG.BLUE,4SPACE}N
     PAB{4SPACE}PQL@{4SPACE}FAQLOQL";               <091>
4410 PRINT"M{8SPACE}@OLJ{4SPACE}PFCD{3SPACE}NOLLLM{5
     SPACE}ALPQQ";                                  <063>
4420 PRINT"J{7SPACE}@OLOB{3SPACE}NLJ{4SPACE}NLLOLF{4
     SPACE}GHIPOLL";                                <221>
4430 PRINT"LJ{5SPACE}IOLOL{4SPACE}LLLJ{4SPACE}FLPO{5
     SPACE}IJPLQOL";                                <006>
4440 PRINT"QLQPOLQLPBAB{4SPACE}LLLL{4SPACE}@PLF{4SPA
     CE}@LLQLLQL";                                  <188>
4450 PRINT"ABFAQOBFF{7SPACE}OPLF{3SPACE}@PLF{4SPACE}
     NLLQQLQQO";                                    <027>
4460 PRINT"CD{2SPACE}AB{9SPACE}NPAB{4SPACE}ABF{5SPAC
     E}IOLQLLQLP";                                  <067>
4470 PRINT"{11SPACE}GH@@PBCD{10SPACE}GILPQLLQQLO";  <025>
4480 PRINT"GH{8SPACE}@IJOPB{13SPACE}IOPLLQQQLQQ";   <077>
4490 PRINT"IJ@{3SPACE,SHIFT-SPACE,SPACE}NLLOLLOJ{2SP
     ACE}GH{2SPACE}@{3SPACE}@IJQLOPOLQQLLL"         <080>
4500 POKE 36614,0:XK(0)=34:YK(0)=196:POKE V+12,35   <140>
4510 POKE V+13,160:POKE V+14,0:POKE V+8,120         <199>
4520 POKE V+9,106:POKE V+10,160:POKE V+11,80        <008>
4530 POKE V+15,0:POKE V+46,0                        <179>
4540 POKE V+46,0:XK(1)=59:YK(1)=56:MB=15:RETURN     <207>
4550 PRINT"{CLR,ORANGE}JCD{2SPACE}ALPQPOQLLPLQLBABAO
     QJ                                             <086>
4560 PRINT"BGH{2SPACE}IQOPLQFEFAB{3SPACE}CD FOQJ{13S
     PACE}I";                                       <244>
4570 PRINT"JIJ@LQPLOQB{3SPACE}CD{7SPACE}E{WHITE}Q{OR
     ANGE}PJ{11SPACE}IB";                           <003>
4580 PRINT"QLOLQLOAB{16SPACE}ALJ{9SPACE}IB ";       <022>
4590 PRINT"LLQOLPFCD{17SPACE}ALJRRRRRRRIB{2SPACE}"; <248>
4600 PRINT"PLQPLB{6SPACE}IJ{5SPACE}IJ@{5SPACE}ALTTTT
     TTTB{3SPACE}";                                 <143>
4610 PRINT"OQPOLJ{6SPACE}ALQBOB{WHITE}Q{ORANGE}FAOJ{
     5SPACE}APJ{3SPACE}IBW{3SPACE}";                <194>
4620 PRINT"LPLQOQ@GH{4SPACE}AOJCD{3SPACE}EOJ{5SPACE}
     QLJ IBØW{3SPACE}";                             <237>
4630 PRINT"POLQPLPIJ{5SPACE}PB{6SPACE}EO{WHITE}Q{ORA
     NGE,4SPACE}ABQPB ØW{3SPACE}";                  <188>
4640 PRINT"LLOLOPLQLJ{4SPACE}AJ{7SPACE}POJ{5SPACE}Q{
     3SPACE}ØW{3SPACE}";                            <082>
4650 PRINT"QLPOQOIOAB{4SPACE}NB{7SPACE}FLO{5SPACE}Q{
     3SPACE}ØW{3SPACE}";                            <107>
4660 PRINT"QPOQOQOBCD{4SPACE}LJ{2SPACE,WHITE}Q{ORANG
     E,4SPACE}@BF{5SPACE}Q{3SPACE}ØW{3SPACE}";      <059>
4670 PRINT"PLLQPLB{6SPACE}AOPQLB{4SPACE}LJ{6SPACE}Q{
     3SPACE}ØW{3SPACE}";                            <131>
4680 PRINT"OQLLOLJ GH{6SPACE}F{6SPACE}AL{4SPACE}GHQ{
     3SPACE}ØW{3SPACE}";                            <047>
4690 PRINT"ABFEFAQJIJ{11SPACE}IJIL{4SPACE}IJLO{2SPAC
     E}ØW{3SPACE}";                                 <243>
4700 PRINT"CD{4SPACE}LPLL@@{8SPACE}@LOOF{4SPACE}OLOP
     J ØW{3SPACE}";                                 <019>
4710 PRINT"{6SPACE}ABPQLOJIJ@{3SPACE}IPQBF{4SPACE}@Q
     OPOLJØW{3SPACE}";                              <081>
4720 PRINT"{6SPACE}CD E{2SPACE}ALLPLQLBAB{5SPACE}@OL
     PLQOLJW{3SPACE}";                              <157>
4730 PRINT"GH{18SPACE}CD{5SPACE}APPOQLPQLJ{3SPACE}"; <190>
4740 PRINT"IJ@{24SPACE}IOLQQLOLQLJ{2SPACE}";        <213>
4750 PRINT"LPQJ@IJGH{16SPACE}GHALOQPOQOLQPJ ";      <131>
4760 PRINT"PLLQLOLIJ@@@@@@@@@@@@@@@@@QIJIQLOOQLPOPLQJ" <077>
4770 POKE 36614,0:XK(0)=43:YK(0)=180:POKE V+12,0    <092>
4780 POKE V+13,0:POKE V+14,145:POKE V+8,216         <223>
4790 POKE V+9,74:POKE V+10,168:POKE V+11,102        <030>
4800 POKE 50175,199:POKE V+23,48:POKE V+29,64       <218>
4810 POKE V+15,123:XK(1)=54:YK(1)=136:MB=15         <162>
4820 ON RI GOTO 4840                                <107>
4830 POKE V+46,1:RETURN                             <022>
4840 FOR Z=9 TO 13:CX=31:CY=Z:GOSUB 3120:PRINT" ":NE
     XT                                             <245>
4850 CX=36:CY=6 :GOSUB 3120:PRINT" "                <028>
4860 CX=36:CY=17:GOSUB 3120:PRINT" "                <088>
4870 FOR Z=7 TO 16:CX=35:CY=Z:GOSUB 3120:PRINT"{2SPA
     CE}"                                           <093>
4880 NEXT:POKE V+46,0:RETURN                        <004>
4890 PRINT"{CLR,YELLOW}OPOLQLLPLLLOQPLOPLLQLRRRRRRR
     LLQLPLLQLOP";                                  <198>
4900 PRINT"POQOLQPOLQLRRRRRRRRRR{WHITE}Q{YELLOW}YYY
     Y{WHITE}Q{YELLOW}RRRRRRRRRRR";                 <213>
4910 PRINT"LQLBF FAOLLBYYYYYYYYYY{6SPACE}YYYYYYYYYYY
     A";                                            <190>
4920 PRINT"OLB{5SPACE}FOQĴ{27SPACE}Ţ";              <145>
4930 PRINT"AB{7SPACE}AOĴ{27SPACE}Ţ";                <235>
4940 PRINT"CD{8SPACE}PĴ{27SPACE}Ţ";                 <185>
4950 PRINT"{10SPACE}RĴ{5SPACE}RRRRR{WHITE}Q{4SPACE}Q
     {YELLOW}RRRRR{6SPACE}Ţ";                       <183>
4960 PRINT"{5SPACE}R{4SPACE}RĴ{3SPACE}R{4SPACE}IJ{6S
     PACE}IJ{4SPACE}R{4SPACE}Ţ";                    <100>
4970 PRINT"GH{3SPACE}R{4SPACE}RĴ{3SPACE}R{2SPACE}RRR
     RRR{2SPACE}RRRRRR{2SPACE}R{4SPACE}Ţ";          <175>
4980 PRINT"IJ@ IR{4SPACE}RĴ{3SPACE}R{2SPACE}RPQQOR{2
     SPACE}RPQQOR{2SPACE}R{4SPACE}Ţ";               <056>
4990 PRINT"POLQLR{4SPACE}RĴ{3SPACE}RJIROQQPRJIROQQPR
     JIR{4SPACE}Ţ";                                 <103>
5000 PRINT"OLOLLR{4SPACE}RĴ{3SPACE}ROPRQOPQROPRQOPQR
     OPR{4SPACE}Ţ";                                 <143>
5010 PRINT"PBCDAR{4SPACE}RĴ{3SPACE}RPORQPOQRPORQPOQR
     POR{4SPACE}Ţ";                                 <113>
5020 PRINT"B{4SPACE}R{4SPACE}RĴ{3SPACE}RBARPQQORBARP
     QQORBAR{4SPACE}Ţ";                             <015>
5030 PRINT"Ĵ{9SPACE}RĴ{3SPACE}R{SPACE,SHIFT-SPACE}RO
     QQPR{2SPACE}ROQQPR{2SPACE}R{4SPACE}Ţ";         <081>
5040 PRINT"Ĵ{9SPACE}RĴ{3SPACE}R{SHIFT-SPACE,SPACE}RR
     RRRR{2SPACE}RRRRRR{2SPACE}R{4SPACE}Ţ";         <105>
5050 PRINT"Ĵ{9SPACE}RĴ{3SPACE}R{4SPACE}AB{6SPACE}AB{
     4SPACE}R{4SPACE}Ţ";                            <001>
5060 PRINT"Ĵ{3SPACE}RRRRRRĴ{5SPACE}RRRRRRR{2SPACE}R
     RRRRRR{6SPACE}Ţ";                              <201>
5070 PRINT"Ĵ{38SPACE}Ţ";                            <100>
5080 PRINT"Ĵ{38SPACE}Ţ";                            <110>
5090 PRINT"J{19SPACE,2SHIFT-SPACE,17SPACE}I";       <224>
5100 PRINT"LQOPOLQOPQOLQOQLPLLLOLLQOPOLLLQOQPLLLQOQ"
     ;                                              <091>
5110 PRINT"{4UP}"                                   <042>
5120 POKE 36614,0:XK(0)=25:YK(0)=96:XK(1)=25        <122>
5130 YK(1)=96:POKE V+12,0:POKE V+13,0               <134>
5140 POKE V+8,240:POKE V+9,60:POKE V+10,201         <070>
5150 POKE V+11,60:POKE 50175,200:POKE V+15,59       <031>
5160 POKE V+23,48+128:POKE V+29,192:MB=0:POKE 33038.
     46                                             <044>
5170 POKE 36610,1:POKE V+14,200:POKE V+21,243:RETURN <239>
5180 REM DATAS     ******************************** <098>
5190 DATA 120,169, 49,141, 20,  3,169,234,141, 21   <102>
5200 DATA  3,169,  0,141, 26,208, 88, 96,120,169    <032>
5210 DATA 51,133,  1,169,  0,133, 95,133, 90,133    <023>
5220 DATA 88,169,208,133, 96,169,240,133, 89,169    <021>
5230 DATA 224,133, 91, 32,191,163,169, 55,133,  1   <150>
5240 DATA 88, 96,5895                               <033>
5250 REM DATEN FUER SPRITES ********************** <095>
5260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:REM 15 X 0  <018>
5270 DATA 64,126,  2, 32,195,  4, 17,129,136, 11    <038>
5280 DATA  0,208,  6,  0, 96, 62,  0,124,122,  0    <086>
5290 DATA 94,119,129,238,120,255, 30, 30, 60,120    <252>
5300 DATA  7,129,224,  0,255,3144                   <042>
5310 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:REM 15 X 0  <068>
5320 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:REM 15 X 0  <078>
5330 DATA  0,  0, 60,  0,  0,126,  0,  0,255,  0    <224>
5340 DATA  1,255,128,  1,255,128,  1,255,128,  0    <103>
5350 DATA 126,0,0,0,0,0,0,0,0,0,0,0,0:REM 13 X 0    <119>
5360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:REM 15 X 0  <118>
5370 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:REM 15 X 0  <128>
5380 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:REM 15 X- 0 <139>
5390 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:REM 15 X 0  <149>
5400 DATA 12,  0, 48, 24,  0, 24, 48,  0, 12,120    <201>
5410 DATA  0, 30,252,  0, 63,192,  0,  3,192,  0,273
     9                                              <215>
5420 DATA  3,192,  0,  3,192,  0,  3,192,  0,  3    <128>
5430 DATA 192,  0,  3,192,  0,  3,192,  0,  3,192   <243>
```

Listing »6510« (Fortsetzung)


64ER ONLINE

```
5440 DATA  0,  3,192,  0,  3,255,255,255,225,108    <204>
5450 DATA 55,225,109,247,239,108,247,227,108,247    <077>
5460 DATA 206,  9,225,206,  8, 33,255,255,255,240   <130>
5470 DATA  0, 15,240,  0, 15,240,  0, 15,  2,  0,668
     9                                              <219>
5480 DATA 64,130,  0, 65,130,  0, 65,194,  0, 67    <143>
5490 DATA 66,  0, 66, 98,  0, 70, 34,  0, 68, 50    <016>
5500 DATA 231, 76, 18,189, 72, 27,129,216,136,129   <235>
5510 DATA 17,207,129,243, 96,129,  6, 63,129,252    <191>
5520 DATA  0,129,  0,255,129,255,  0,129,  0,  3    <179>
5530 DATA 129,192, 14,129,112, 56,129, 28,224,255,57
     65                                             <051>
5540 DATA  7,  0,  0,248,  0,  1,252,  0,  3,  6    <144>
5550 DATA  0,  6,  3,  0,  6,  3,  0,  6,  3,  0    <205>
5560 DATA  6,  3,  0,  6,  3,  0,  3,  6,  0,  3    <218>
5570 DATA 252,  0,  7,248,  0, 15,  0,  3,158,  0   <075>
5580 DATA  3,252,  0,  9,248,  0,125,240,  0, 63    <186>
5590 DATA 224,  0, 63,192,  0, 31,128,  0, 15,  0,28
     43                                             <185>
5600 DATA  0, 14,  0,  0,  0,  0,  0,  3,195,192    <247>
5610 DATA 15,126,240, 15,255,240,  3,  0,192,  3    <109>
5620 DATA 126,192, 15, 66,240, 15,126,240,  3,  0   <173>
5630 DATA 192,  3,126,192, 15,  0,240, 15, 94,240   <187>
5640 DATA  3, 82,192,  3,114,192, 15,  0,240, 15,448
     9                                              <098>
5650 DATA 78,240,  3, 74,192,  3,126,192, 15,  0    <116>
5660 DATA 240, 15,255,240,  3,255,192               <086>
5670 DATA 200,193,210,193,204,196, 32,194,197,201   <023>
5680 DATA 206,197, 32, 38, 32,193,210,206,197, 32   <145>
5690 DATA 202,193,206,211,197,206                   <069>
5700 REM DATEN FUER DIE ZEICHEN *****************   <041>
5710 DATA  8,  8, 28, 28, 62,126,127,255,127, 63    <241>
5720 DATA 31, 15,  3,  7,  7,  3,255,254,252,240    <129>
5730 DATA 248,240,240,224,  0,  1,  1,  0,  0,  0   <011>
5740 DATA  0,  0,224,128,192,128,128,  0,  0,  0    <134>
5750 DATA 252,110, 60, 24, 16,  0,  0,  0,252,126,10
     916                                            <232>
5760 DATA 124, 56, 56, 16, 16, 16,  0,  0,  0,  1   <008>
5770 DATA  1,  3,  3,  7,  0,  0,  0,  0,  0,128    <008>
5780 DATA 128,192,  4, 15, 15, 31, 31, 63, 63,255   <038>
5790 DATA 192,224,224,240,240,248,252,255,  8,  8   <254>
5800 DATA  8, 28, 28, 62,126,255, 46,127,223,126    <170>
5810 DATA 239,251,191,237,192,224,112,248,252, 76   <176>
5820 DATA 76,196,  3,  6, 14, 31, 63, 50, 51, 34    <187>
5830 DATA 63, 31,143,207,199,225,240,252,252,248    <139>
5840 DATA 225,195,207,143,143, 63,255,255,255,255,10
     422                                            <244>
5850 DATA 255,255,255,255,  0,126,126,102,102,126,12
     6,0                                            <177>
5860 REM ** MASCHINENROUTINE **                     <237>
5870 DATA 120,169, 32,141, 20,  3,169,128,141       <123>
5880 DATA 21,  3,169,129,141, 26,208,173,254        <145>
5890 DATA 143,141, 18,208,173, 17,208, 41,127       <201>
5900 DATA 141, 17,208, 88, 96,173, 25,208,141       <175>
5910 DATA 25,208, 48, 10, 32, 95,128,173, 13,6425   <072>
5920 DATA 220, 88, 76, 49,234,173,253,143,201       <209>
5930 DATA  1,240,19,169,  0,141, 32,208,173         <085>
5940 DATA 254,143,141, 18,208,169,  1,141,253       <251>
5950 DATA 143, 76,188,254,173,252,143,141, 32       <064>
5960 DATA 208,169,255,141, 18,208,169,  2,141       <025>
5970 DATA 253,143, 76,188,254,173,  1,220, 41,7590  <237>
5980 DATA  1,208,101,173,  1,220, 41,  2,208
                                                    <068>
5990 DATA 91,173,  1,220, 41,  4,208, 78,173        <103>
6000 DATA  1,220, 41,  8,208, 74,165,248,141        <161>
6010 DATA  0,208,141,  2,208,141,  4,208,141        <154>
6020 DATA  6,208,165,249,141,  1,208,141,  3        <179>
6030 DATA 208,141,  5,208,141,  7,208,173,  6,6321  <183>
6040 DATA 143,141, 16,208,206,  0,143,240, 55       <033>
6050 DATA 206,  1,143,240, 92,206,  5,143,240       <249>
6060 DATA 34,173,  2,143,208, 92,173, 30,208        <015>
6070 DATA 234,234, 32, 21,129,234,234,234,234       <170>
6080 DATA 234, 32,125,129, 96, 76, 82,129, 76       <059>
6090 DATA 89,129, 76, 77,129, 76, 72,129,238        <087>
6100 DATA 254,143,169,100,141,  5,143, 76,172,8131  <149>
6110 DATA 128,173,  3,143, 73,  1,141,  3,143       <213>
6120 DATA 208, 16,173, 21,208, 41,207,141, 21       <114>
6130 DATA 208,169, 90,141,  0,143, 76,162,128       <142>
6140 DATA 173, 21,208,  9, 48,141,21,208,169        <102>
6150 DATA 45,141,  0,143, 76,162,128,169,150        <159>
6160 DATA 141,  1,143,234,234, 76,167,128,238,6207  <218>
6170 DATA 40,208, 76,177,128, 41,  8,165,248        <145>
6180 DATA 201, 24,240, 20,201, 64,208, 34,173       <169>
6190 DATA 16,208, 41,  3,240, 27,198,248,165        <157>
6200 DATA 250,208, 21, 76, 60,129,173, 16,208       <210>
6210 DATA 41,  3,208, 11,230,248,165,250,240        <205>
6220 DATA  5,169,  1,133,251,234, 96,169,  3,6960   <088>
6230 DATA 141,255,127, 96,230,249, 76,102,128       <091>
6240 DATA 198,249, 76,109,128,230,248,240, 14       <108>
6250 DATA 76,116,128,198,248,165,248,201,255        <174>
6260 DATA 240, 14, 76,123,128,173,  6,143,  9       <170>
6270 DATA 15,141,  6,143, 76,116,128,173,  6        <180>
6280 DATA 143, 41,240,141,  6,143, 76,123,128,7147  <023>
6290 DATA 234,234,234,206,255,142,240,  1, 96       <089>
6300 DATA 169,  7,141,255,142,238,254,142,174       <162>
6310 DATA 254,142,189,  0,197,141,  0,212,189       <067>
6320 DATA  0,198, 41,127,141,  1,212,189,  0        <170>
6330 DATA 198, 41,128,208,  6,169, 64,141,  4       <251>
6340 DATA 212, 96,169, 64,141,  4,212,169, 65,7254  <053>
6350 DATA 141,  4,212, 96,120,169,128,141, 42       <097>
6360 DATA 128,169, 95,141, 41,128, 88, 96,120       <132>
6370 DATA 169,129,141, 42,128,169,125,141, 41       <227>
6380 DATA 128, 88, 96                               <089>
6390 REM    T  A  K  T    1 ************            <250>
6400 DATA 27,132, 27,  4, 55,136, 39,134,3869       <122>
6410 DATA 27,132, 27,  4, 55,136, 39,134            <126>
6420 DATA 27,132, 27,  4, 55,136, 39,134            <136>
6430 DATA 27,132,226,132, 39,134, 55,136            <037>
6440 REM    T  A  K  T    2 ************            <046>
6450 DATA 27,132, 27,  4, 55,136, 39,134            <166>
6460 DATA 27,132, 27,  4, 55,136, 39,134,3070       <166>
6470 DATA 27,132, 27,  4, 55,136, 39,134            <186>
6480 DATA 27,132,226,132, 39,134, 55,136            <087>
6490 REM    T  A  K  T    3 ************            <097>
6500 DATA 27,132, 27,  4, 55,136, 39,134            <216>
6510 DATA 27,132,226,132, 39,134, 55,136            <117>
6520 DATA 169,131,169,  3, 81,135,123,133,3787      <132>
6530 DATA 169,131,156,132,123,133, 81,135           <232>
6540 REM    T  A  K  T    4 ************            <148>
6550 DATA 66,131, 66,  3,133,134,226,132            <101>
6560 DATA 66,131, 81,135,226,132,133,134            <210>
6570 DATA 27,132, 27,  4, 55,136, 39,134            <030>
6580 DATA 27,132,226,132, 39,134, 55,136,4255       <183>
6590 REM    T  A  K  T    5 ************            <199>
6600 DATA 27,132, 27,  4, 55,136, 39,134            <060>
6610 DATA 27,132, 27,  4, 55,136, 39,134            <070>
6620 DATA 27,132, 27,  4, 55,136, 39,134            <080>
6630 DATA 27,132,276,132, 39,134, 55,136            <237>
6640 REM    T  A  K  T    6 ************            <250>
6650 DATA 27,132, 27,  4, 55,136, 39,134            <110>
6660 DATA 27,132, 27,  4, 55,136, 39,134,3624       <116>
6670 DATA 27,132, 27,  4, 55,136, 39,134            <131>
6680 DATA 27,132,226,132, 39,134, 55,136            <032>
6690 REM    T  A  K  T    7 ************            <046>
6700 DATA 27,132, 27,  4, 55,136, 39,134            <161>
6710 DATA 27,132,226,132, 39,134, 55,136            <062>
6720 DATA 169,131,169,  3, 81,135,123,133           <072>
6730 DATA 169,131,156,132,123,133, 81,135           <177>
6740 REM    T  A  K  T    8 ************            <097>
6750 DATA 66,131, 66,  3,133,134,226,132,5738       <049>
6760 DATA 66,131, 81,135,226,132,133,134            <155>
6770 DATA 66,131, 66,  3,133,134,226,132            <066>
6780 DATA 66,131, 81,135,226,132,133,134            <175>
6790 REM    T  A  K  T    9 ************            <148>
6800 DATA 226,132,226,  4,196,137, 81,135,4038      <151>
6810 DATA 226,132,226,  4,196,137, 81,135           <166>
6820 DATA 226,132,226,  4,196,137, 81,135           <176>
6830 DATA 226,132, 39,134, 81,135,196,137           <240>
6840 REM    T  A  K  T   10 ************            <238>
6850 DATA 226,132,226,  4,196,137, 81,135           <206>
6860 DATA 226,132,226,  4,196,137, 81,135,5402      <207>
6870 DATA 226,132,226,  4,196,137, 81,135           <226>
6880 DATA 226,132, 39,134, 81,135,196,137           <034>
6890 REM    T  A  K  T   11 ************            <033>
6900 DATA 39,134, 39,  6, 78,140, 56,137            <117>
6910 DATA 039,134,081,135,056,137,078,140           <159>
6920 DATA 123,133,123,  5,247,138, 55,136,4380      <010>
6930 DATA 123,133,232,134,055,136,247,138           <173>
6940 REM    T  A  K  T   12 ************            <085>
6950 DATA 226,132,226,  4,196,137, 81,135           <051>
6960 DATA 226,132, 39,134, 81,135,196,137           <115>
6970 DATA 226,132,226,  4,196,137, 81,135           <071>
6980 DATA 226,132, 39,134, 81,135,196,137           <135>
6990 REM    T  A  K  T   13 ************            <136>
7000 DATA 226,132,226,  4,196,137, 81,135,6646      <103>
7010 DATA 226,132,226,  4,196,137, 81,135           <111>
7020 DATA 226,132,226,  4,196,137, 81,135           <121>
7030 DATA 226,132, 39,134, 81,135,196,137           <185>
7040 REM    T  A  K  T   14 ************            <187>
7050 DATA 226,132,226,  4,196,137, 81,135           <151>
7060 DATA 226,132,226,  4,196,137, 81,135,5402      <152>
7070 DATA 226,132,226,  4,196,137, 81,135           <171>
7080 DATA 226,132, 39,134, 81,135,196,137           <235>
7090 REM    T  A  K  T   15 ************            <238>
7100 DATA 39,134, 39,  6, 78,140, 56,137            <062>
7110 DATA 39,134, 81,135, 56,137, 78,140            <168>
7120 DATA 123,133,123,  5,247,138, 55,136           <215>
7130 DATA 123,133,232,134, 55,136,247,138,5578      <074>
7140 REM    T  A  K  T   16 ************            <033>
7150 DATA 226,132,226,  4,196,137, 81,135           <251>
7160 DATA 226,132, 39,134, 81,135,196,137           <059>
7170 DATA 27,132, 27,  4, 55,136, 39,134            <121>
7180 DATA 27,132,226;132, 39,134, 55,136            <022>
7190 :                                              <108>
7200 DATA 120,169, 50,141,  8,212,141,  1,212,169,46
     49                                             <037>
7210 DATA  3,141,  3,212,169, 17,141, 12,212,141    <018>
7220 DATA  5,212,169, 32,141, 11,212,169, 33,141    <087>
7230 DATA 11,212,169,242,141,  6,212,169,114,141    <197>
7240 DATA 13,212,169, 15,141, 24,212,169, 16,141    <160>
7250 DATA  4,212,169,129,141,  4,212,173, 32,208    <123>
7260 DATA 72,173, 33,208, 72,162,  2,134,  2,162,700
     6                                              <027>
7270 DATA 128,160,255,238, 32,208,238, 33,208,173   <051>
7280 DATA 24,212, 73, 15,141, 24,212,136,208,239    <201>
7290 DATA 202,208,234,198,  2,208,228, 88,169,  0   <182>
7300 DATA 141,  4,212,141, 11,212,141, 24,212,104   <193>
7310 DATA 141, 33,208,104,141, 32,208, 96           <144>
7320 DATA 169,  4,141,136,  2,169,131,141,  2,  3   <041>
7330 DATA 169,164,141,  3,  3,  0,  0,172,  1,  0   <095>
7340 DATA  2, 80,130,  1,  1,  3,172,110, -1,  0    <084>
7350 DATA  2, 42, 42,42, 32,28, 54, 53, 49, 48      <190>
7360 DATA 32,158, 42, 42, 42,254,172, -1,  0,  2    <021>
7370 DATA 130, 80, -1,  1,  3,173,255,  1,  0,  2   <125>
7380 DATA  0, 70                                    <054>
```

Listing »6510« (Schluß)

Ordnung ist das halbe Leben

Mit diesem Programm bringen Sie auf bequeme Weise Ordnung in die Directories Ihrer Disketten.

Ist Ihnen das auch schon mehrmals passiert: Sie hatten auf einer Diskette ein schön geordnetes Directory und beim Abspeichern eines weiteren Files stand dieses nicht sauber am Schluß des Directory, sondern mitten zwischen den anderen Programmen. Mit dem hier vorgestellten Directory-Sortierprogramm ist es nun möglich, diese »falsch hineingeratenen« Files herauszunehmen und an geeigneter Stelle wieder einzusetzen. Auch das Einfügen von Trennstrichen ist möglich. Bild 1 und Bild 2 zeigt Ihnen den Unterschied eines unsortierten zu einem sortierten Directory.

Nach dem Starten des Programms legt man die Diskette ein, die sortiert werden soll, und drückt eine Taste. In der linken oberen Ecke wird nun die Sektornummer des Directory-Blocks angezeigt, den der Computer gerade einliest, rechts daneben die Anzahl der in den Speicher eingelesenen Files. Nach beendetem Einlesen erscheint links vom obersten File ein schwarzer Pfeil, der sich mit den Funktionstasten F3 nach oben und F5 nach unten bewegen läßt. Mittels F1 kann nun ein mit diesem Pfeil gekennzeichnetes File nach rechts herausgeschoben und mit F3 und F5 verschoben werden. An gewünschter Stelle wird es mit F1 wieder eingefügt. Somit ist ein beliebiges Vertauschen aller Files im Directory möglich.

F8 schreibt das sortierte Directory wieder auf die Diskette zurück. Will man das gänderte Directory nicht abgespeichert haben, so kann statt dessen mit F6 noch einmal das alte oder ein anderes eingelesen werden.

Zur optischen Abgrenzung mehrerer Files dient der Trennstrich, den man mit F2 erzeugen, mit F3 und F5 verschieben und schließlich wieder mit F1 einfügen kann. Wem übrigens ein anderer Trennstrich besser gefällt, der kann in Zeile 500 für die Minuszeichen andere einsetzen, zum Beispiel SHIFT+*. SHIFT + C ist nicht zu empfehlen, da dieser Strich bei Groß-Kleinschrift-Umschaltung ein großes »C« ergibt. Der Trennstrich belegt keinen Block auf der Diskette und ist zur besseren Unterscheidung mit DEL im Directory gekennzeichnet. Ein versehentlich mit »F2« erzeugter Trennstrich kann, wenn er nach rechts gebracht wird, mit F4 wieder gelöscht werden. In gleicher Weise können auch Files aus dem Directory gelöscht werden. Dabei werden aber die von dem File belegten Blöcke nicht wieder freigegeben, so daß ein abschließendes »VALIDATE« der Diskette erforderlich ist, wenn man ein File gelöscht hat.

Nun zum Aufbau des Programms. Es ist vorwiegend in Basic geschrieben und enthält zwei kurze Maschinensprache-Routinen. Die erste dient zum Einlesen der Filenamen und Fileparameter und wird mit SYS 52992, log. Filenummer, Länge, String aufgerufen. Das Lesen mit GET# in Basic wäre dafür viel zu langsam. Mit dem zweiten Maschinenprogramm werden die am Bildschirm angezeigten Files nach oben oder unten gescrollt. Aufgerufen wird es mit SYS 53056,r,oz,ls,uz,rs.

Dabei haben die Parameter folgende Bedeutung:

r = Scrollrichtungen: 0 für nach oben und 1 für nach unten
oz = oberste Zeile,
ls = linke Spalte,
uz = unterste Zeile,
rs = rechte Spalte.

Durch Abspeichern des Speicherbereichs von $CF00 bis $CFFF kann diese Routine auch in andere Basic- oder Maschinenprogramme eingebunden werden. (Edwin Göbel/hm)

```
0 [illegible]
23    "MORSETRAINER"      PRG
2     "DEMO"              SEQ
23    "SUPERVOC"          PRG
40    "LEHRERKALENDER"    PRG
39    "CHART-GRAFIK(SB)"  PRG
27    "HOT WHEELS"        PRG
17    "LOGIC DISASSEMBL"  PRG
7     "MASCHINE G.MANOR"  PRG
3     "SUPERVOC.DEMO"     PRG
6     "GRAFIK GHOST M."   PRG
13    "MOVEMASTER"        PRG
10    "GHOST MANOR"       PRG
53    "UNDERGROUND"       PRG
7     "LOADER MOVEM."     PRG
393 BLOCKS FREE.

READY.
```

Bild 1. Das Durcheinander eines unsortierten Directorys

```
0 [illegible]
23    "MORSETRAINER"      PRG
0     "----------------"  DEL
40    "LEHRERKALENDER"    PRG
2     "DEMO"              SEQ
0     "----------------"  DEL
23    "SUPERVOC"          PRG
3     "SUPERVOC.DEMO"     PRG
0     "----------------"  DEL
39    "CHART-GRAFIK(SB)"  PRG
0     "----------------"  DEL
27    "HOT WHEELS"        PRG
0     "----------------"  DEL
17    "LOGIC DISASSEMBL"  PRG
0     "----------------"  DEL
7     "MASCHINE G.MANOR"  PRG
6     "GRAFIK GHOST M."   PRG
10    "GHOST MANOR"       PRG
0     "----------------"  DEL
7     "LOADER MOVEM."     PRG
13    "MOVEMASTER"        PRG
0     "----------------"  DEL
53    "UNDERGROUND"       PRG
393 BLOCKS FREE.

READY.
```

Bild 2. Der Directory-Sorter schafft schnell Ordnung

```
100 REM ************************             <227>
110 REM *                      *             <081>
120 REM * DIRECTORY-SORT V1.0  *             <106>
130 REM * -------------------  *             <188>
140 REM * AUTOR:               *             <052>
150 REM * EDWIN GOEBEL         *             <158>
160 REM * SCHUETZSTR. 1        *             <001>
170 REM * 8522 HERZOGENAURACH  *             <116>
180 REM *                      *             <166>
190 REM ************************             <061>
200 GOTO 430                                 <232>
210 REM ********************************     <119>
```

Listing zum Directory-Sorter.
Beachten Sie den Checksummer auf Seite 54.

```
220 REM  UNTERROUTINEN                          <109>
230 REM ********************************        <139>
240 INPUT#1,F1,F$,F2,F3:IF F1=0 THEN RETUR
    N                                           <122>
250 PRINT"{CYAN,DOWN}"F1;F$,F2;F3"{BLACK}"
    :END:REM "{CYAN}"=CYAN; "{BLACK}"=SCHW
    ARZ                                         <011>
260 REM --------------------------------        <007>
270 IF NO+EN>=C THEN RETURN                     <079>
280 SYS 53056,0,2,2,23,18:NO=NO+1               <249>
290 POKE 214,23:POKE 211,2:SYS 58732:PRINT
    NA$(ZU(NO+EN-1));:RETURN                    <022>
300 REM --------------------------------        <047>
310 IF NO=0 THEN RETURN                         <118>
320 SYS 53056,1,2,2,23,18:NO=NO-1               <035>
330 POKE 214,2:POKE 211,2:SYS 58732:PRINT
    NA$(ZU(NO+1)):RETURN                        <146>
340 REM --------------------------------        <087>
350 PRINT"{CLR,BLACK,RVSON}"TAB(13)"DIRECT
    ORY-SORT"                                   <015>
360 PRINT TAB(12)"---------------"              <069>
370 PRINT"{CYAN,DOWN,SPACE}DISKETTE EINLEG
    EN UND TASTE DRUECKEN"                      <093>
380 GET T$:IF T$=""THEN 380                     <011>
390 PRINT"{HOME,3DOWN,37SPACE}":RETURN          <114>
400 REM ********************************        <054>
410 REM  BEGINN DES HAUPTPROGRAMMES             <056>
420 REM ********************************        <074>
430 POKE 53280,0:POKE 53281,11:GOSUB 350        <101>
440 OPEN 1,8,15,"I":DIM AN$(145),NA$(145),
    RE$(145),ZU(146),SN(19)                     <129>
450 FOR I=1 TO 18:READ SN(I):NEXT               <053>
460 FOR I=52992 TO 53242:READ S:POKE I,S:N
    EXT                                         <097>
470 GOSUB 240:OPEN 2,8,2,"#":GOSUB 240          <183>
480 S=1:C=1:N$=CHR$(0):NN$=N$+N$+N$+N$+N$+
    N$+N$+N$+N$+N$                              <055>
490 NU$=NN$+NN$+NN$:RE$(0)=NN$+N$               <252>
500 AN$(0)=CHR$(128)+CHR$(18)+CHR$(1):NA$(
    0)="----------------":REM TRENNSTRICH       <253>
510 REM ********************************        <164>
520 REM  EINLESEN DES DIRECTORYS                <208>
530 REM ********************************        <185>
540 PRINT#1,"U1 2 0 18"S:PRINT"{HOME,LIG.R
    ED}"S"{LEFT,SPACE}":REM "{LIG.RED}"=HE
    LLROT                                       <233>
550 GET#2,T$:GET#2,S$:S=ASC(S$+N$)              <126>
560 FOR BP=0 TO 7:PRINT#1,"B-P 2";BP*32+2       <032>
570 SYS 52992,2,3,X$:AN$(C)=X$:IF LEFT$(X$
    ,1)=N$THEN NEXT:GOTO 600                    <086>
580 SYS 52992,2,16,X$:NA$(C)=X$:SYS 52992,
    2,11,X$:RE$(C)=X$                           <101>
590 ZU(C)=C:PRINT"{HOME,4RIGHT}"C"{LEFT,SP
    ACE}":C=C+1:NEXT                            <083>
600 IF T$<>""THEN 540                           <069>
610 CLOSE 2:POKE 650,128:PRINT"{HOME,2DOWN
    ,BLACK}>{CYAN,UP}";:CP=2:NO=0:NU=0:EN=
    C:IF EN>23 THEN EN=23                       <152>
620 PRINT CHR$(13)TAB(2)NA$(NU+1);:NU=NU+1
    :IF NU<EN-1 THEN 620                        <020>
630 REM ********************************        <029>
640 REM  TASTATURABFRAGE                        <113>
690 REM ********************************        <089>
700 GET TA$:IF TA$=""THEN 700                   <202>
710 IF TA$="{F3}"THEN IF CP>2 THEN CP=CP-1
    :SYS 53056,0,2,0,23,0                       <105>
720 IF TA$="{F3}"THEN IF CP=2 THEN GOSUB 3
    10                                          <003>
730 IF TA$="{F5}"THEN IF CP<EN THEN IF NO+
    CP<C THEN CP=CP+1:SYS 53056,1,2,0,23,0      <227>
740 IF TA$="{F5}"THEN IF CP>=23 THEN GOSUB
    270                                         <001>
750 IF TA$="{F1}"THEN 810                       <239>
760 IF TA$="{F2}"THEN IF C<145 THEN F=1:C=
    C+1:PRINT"{LIG.RED,HOME,4RIGHT}"C-1"{L
    EFT,SPACE,CYAN}":EN=EN+1:IF EN>23 THEN
    EN=23                                       <027>
770 IF TA$="{F2}"THEN IF F=1 THEN F=0:TE=0
    :TE$=NA$(0):GOTO 840                        <227>
780 IF TA$="{F8}"THEN 1000                      <061>
790 IF TA$="{F6}"THEN 1090                      <079>
800 GOTO 700                                    <035>
810 TE=ZU(NO+CP-1):TE$=NA$(TE)                  <116>
820 SYS 53056,0,CP,2,23,18:IF C>23 THEN EN
    =EN+1:GOSUB 290:EN=EN-1                     <063>
830 FOR I=NO+CP-2 TO C-1:ZU(I+1)=ZU(I+2):N
    EXT                                         <040>
840 POKE 214,CP:POKE 211,19:SYS 58732           <020>
850 PRINT"TT"TE$:REM "T"=COMMODORE+"T"          <138>
860 GET TA$:IF TA$=""THEN 860                   <114>
870 IF TA$="{F3}"THEN IF CP>2 THEN CP=CP-1
    :SYS 53056,0,2,19,24,37:SYS 53056,0,2,
    0,23,0                                      <093>
880 IF TA$="{F3}"THEN IF CP=2 THEN GOSUB 3
    10                                          <164>
890 IF TA$="{F5}"THEN IF CP<EN THEN CP=CP+
    1:SYS 53056,1,2,19,24,37:SYS 53056,1,2
    ,0,23,0                                     <214>
900 IF TA$="{F5}"THEN IF CP=23 THEN GOSUB
    270                                         <241>
910 IF TA$="{F1}"THEN 950                       <149>
920 IF TA$="{F4}"THEN TE$="{16RIGHT}":C=C-
    1:EN=C:IF EN>23 THEN EN=23                  <088>
930 IF TA$="{F4}"THEN PRINT"{HOME,LIG.RED,
    4RIGHT}"C-1"{LEFT,SPACE,CYAN}":GOTO 97
    0                                           <012>
940 GOTO 860                                    <214>
950 SYS 53056,1,CP,2,23,18                      <250>
960 FOR I=C-1 TO NO+CP-1 STEP-1:ZU(I+1)=ZU
    (I):NEXT:ZU(NO+CP-1)=TE                     <141>
970 POKE 214,CP:POKE 211,2:SYS 58732            <094>
980 PRINT TE$"{19SPACE}"                        <113>
990 GOTO 700                                    <001>
1000 REM *******************************        <102>
1010 REM  SCHREIBEN DES SORTIERTEN DIR.         <015>
1020 REM *******************************        <122>
1030 OPEN 2,8,2,"#":T=18:FOR I=0 TO INT((C
     -2)/8):IF 8*I+8>=C-1 THEN T=0              <242>
1040 PRINT#1,"B-P 2 0":PRINT#2,CHR$(T)CHR$
     (SN(I+2))                                  <057>
1050 FOR BP=0 TO 7:PRINT#1,"B-P 2";BP*32+2      <012>
1060 IF BP+8*I+1>=C THEN PRINT#2,NU$;:NEXT
     :GOTO 1080                                 <132>
1070 PRINT#2,AN$(ZU(BP+8*I+1))NA$(ZU(BP+8*
     I+1))RE$(ZU(BP+8*I+1));:NEXT               <255>
1080 PRINT#1,"U2 2 0 18"SN(I+1):PRINT"{LIG
     .RED,HOME}"TAB(36)SN(I+1)"{LEFT,SPACE
     ,CYAN}":GOSUB 240:NEXT                     <129>
1090 CLOSE 2:GOSUB 350:GOTO 470                 <213>
1100 REM *******************************        <203>
1110 REM  SEKTOREN DES DIRECTORY TRACKS         <173>
1120 REM *******************************        <223>
1130 DATA 1,4,7,10,13,16,2,5,8,11,14,17,3,
     6,9,12,15,18                               <071>
1140 REM *******************************        <243>
1150 REM  INPUT MASCHINEN-ROUTINE               <138>
1160 REM *******************************        <007>
1170 DATA 32,253,174,32,158,183,32,30,225,
     32,253,174,32,158,183,138,72,32,253        <010>
1180 DATA 174,32,139,176,133,73,132,74,32,
     163,182,104,32,117,180,160,2,185           <122>
1190 DATA 97,0,145,73,136,16,248,200,32,18
     ,225,145,98,200,196,97,208,246,32          <201>
1200 DATA 204,255,96,0,0,0,0,0                  <252>
1210 REM *******************************        <057>
1220 REM  SCROLL MASCHINEN-ROUTINE              <015>
1230 REM *******************************        <077>
1240 DATA 32,245,207,138,72,32,245,207,224
     ,0,176,3,76                                <172>
1250 DATA 72,178,224,24,176,249,134,251,32
     ,245,207,224,0,144,240,224,39,176          <251>
1260 DATA 236,134,253,32,245,207,224,25,17
     6,227,134,252,232,138,56,229,251           <213>
1270 DATA 144,218,240,216,133,250,32,245,2
     07,224,40,176,207,228,253,144,203          <254>
1280 DATA 134,254,104,170,165,172,72,165,1
     73,72,165,174,72,165,175,72,224,0          <030>
1290 DATA 208,22,166,251,198,250,240,44,32
     ,240,233,232,189,240,236,133,172           <234>
1300 DATA 181,217,32,219,207,48,236,202,24
     0,3,76,72,178,166,252,198,250,240          <053>
1310 DATA 16,32,240,233,202,189,240,236,13
     3,172,181,217,32,219,207,48,236,164        <148>
1320 DATA 254,32,240,233,32,36,234,169,32,
     145,209,136,196,253,16,249,76,88           <034>
1330 DATA 233,41,3,13,136,2,133,173,32,224
     ,233,164,254,177,172,145,209,177           <016>
1340 DATA 174,145,243,136,196,253,16,243,9
     6,32,253,174,76,158,183                    <128>
```

Listing zum Directory-Sorter (Schluß)

Basic-Befehle im Griff

Gerade für den Anfänger ist es nicht immer leicht, sich alle Basic-Befehle zu merken. Man denke nur an ausgefallene Befehle wie CMD oder POS(X).

Exbasic Level II kennt als Abhilfe den Befehl HELP, mit dem eine Liste sämtlicher Exbasicbefehle auf den Bildschirm gebracht werden kann. Die folgenden kurzen Programme (drei Lösungsmöglichkeiten für dasselbe Problem), die sich als Hilfsprogramme (Utility) zum Einbau in ein zu entwickelndes längeres Hauptprogramm verstehen, simulieren diesen Befehl HELP für das reine Basic, das heißt, sie geben eine Liste aller Befehle auf dem Bildschirm aus.

Wie geschieht das nun im einzelnen?

Eine unmittelbare Ausgabe aller Basic-Befehle auf dem Bildschirm per PRINT-Anweisungen hätte zwar gegenüber den neuen zu besprechenden drei Methoden den Vorteil, daß man diese Liste alphabetisch ordnen könnte. Sie wäre aber viel zu langsam (in der Größenordnung 5 Sekunden) und das generierte Basic-Programm wäre viel zu lang: Eine Liste aller aneinandergereihten Basic-Befehle, mit Trennzeichen zwischen Befehl und Befehl, umfaßt 331 Bytes. Einschließlich PRINT-Befehle, Hochkommata und Zeilennumerierungen würde ein solches Programm also mindestens 375 Bytes benötigen.

Nun enthält aber das Betriebssystem bereits eine Tabelle aller Basic-Befehle: 49310-49565 ($C09E-$C19D). Diese hat lediglich den Nachteil, daß sie keine Trennzeichen verwendet, sondern das Ende eines Basic-Befehlswortes dadurch kennzeichnet, daß der ASCII-Code des betreffenden Zeichens um den Wert 128 (Bit 7 gesetzt) erhöht wird. Das Vorhandensein einer solchen Tabelle wollen wir in den folgenden drei Vorschlägen zur Simulation von HELP ausnützen.

Methode 1: »HELP« in Basic

Das Basic-Programm nach Listing 1 ist der kürzeste unserer drei Vorschläge (nur 80 Bytes lang), es hat aber den Nachteil, daß es zum Aufbau der Basic-Befehlsliste mehr also vier Sekunden benötigt. Es wird (im Direktmodus oder vom Hauptprogramm aus) per GOSUB500 aufgerufen und PRINTet Zeichen für Zeichen der Tabelle 49310 — 49565, wobei es darauf achtet, daß immer wenn ein Befehlswort zu Ende ist, das betreffende Zeichen vor dem Ausdrucken (auf dem Bildschirm) zuerst normalisiert wird und daß dann zusätzlich noch ein Befehlsworttrennzeichen eingefügt wird (in Listing 1 an der Stelle X$=" . ", welches Symbol vom Leser beliebig abgeändert und individuellem Geschmack angepaßt werden kann). Nachteilig ist bei allen drei zu besprechenden Vorschlägen, daß man sich sinnvollerweise mit der durch die CBM-Tabelle vorgegebenen Unordnung zufriedengeben muß.

```
500 REM:HELP
510 FORI=0TO254:X=PEEK(49310+I):IFX>99TH
ENX=X-128:X$=".":GOTO530
520 X$=""
530 PRINTCHR$(X)+X$;:NEXT:RETURN
READY.
```

Listing 1. Simulation von HELP als reines Basicprogramm, Länge 80 Bytes, Ausführungszeit 4 s

Methode 2: »HELP« per Maschinenprogramm

Das in Listing 2 aufgeführte Maschinenprogramm zur Ausgabe der Basic-Befehlsliste auf dem Bildschirm ist zwar ein wenig länger als das Basic-Programm in Listing 1, nämlich 106 Bytes lang. Dafür benötigt es aber zum Einlesen der Data-Zeilen deutlich weniger als ⅓ Sekunde und erzeugt die Basic-Befehlsliste nach dem Aufruf per SYS700 in einem kaum wahrnehmbaren Bruchteil einer Sekunde. Das eigentliche Maschinenprogramm ist nur 21 Bytes lang und kann (voll verschiebbar) überall abgelegt werden, wo es nicht stört, zum Beispiel in einem durch Herabsetzen der RAM-Grenzen geschützten Maschinensprachbereich oder im Kassettenpuffer oder eben auch dort, wo wir es hingelegt haben, nämlich in den für den Benutzer freien Bereich 678-767 (02A6-02FF). Wir haben es als mit SYS700 aufrufbar gestaltet. Bei Ablage an anderer Stelle aaaa müßte POKE700+I,X in Zeile 610 von Listing 2 durch POKEaaaa+I,X ersetzt und das Programm per SYSaaaa aufgerufen werden. Legt man das Programm nicht in den Kassettenpuffer, sondern wie hier vorgeschlagen, nach 700ff., dann kann man das generierende Basic-Programm aus Listing 2 natürlich nach dem Einlesen des Maschinenprogramms wieder löschen (ein Maschinenprogramm für größere Löschvorhaben wurde im 64'er Ausgabe 5/84, Seite 85, veröffentlicht). Listing 3 gibt eine ausführlich kommentierte Darstellung des Maschinenprogramms in Assemblernotierung.

Methode 3: Maschinenprogrammerzeuger ohne Datenzeilen

Will man das Maschinenprogramm in den Kassettenpuffer legen, so muß man es gegebenenfalls (bei zwischenzeitlicher Ausführung von LOAD, SAVE oder VERIFY) oder, wenn sich

```
600 REM:HELP
610 FORI=0TO20:READX:POKE700+I,X:NEXT
620 DATA 160,255,200,185,158,192,16,7,41
,127,32,71,203,169,46,32,71,203,208,238,
96
READY.
```

Listing 2. Simulation von HELP per Maschinenprogrammlader. Länge 106 Bytes. Einlesen 0,3 Sekunden. Ausführung »augenblicklich«, Ansprung SYS700.

mehrere Hilfsprogramme den Kassettenpuffer teilen) vor jedem Aufruf neu generieren. Die Schwierigkeiten, die dadurch entstehen, daß der Basic-Datenzeiger per RESTORE nur immer an den Datenanfang gesetzt werden kann, lassen sich beispielsweise durch ein in Computer persönlich, Ausgabe 10/84, Seite 52, beschriebenes Hilfsprogramm beseitigen, mit welchem der Datenzeiger an beliebige Stellen gesetzt werden kann. Das nun noch zu beschreibende Programm nach Listing 4 vermeidet die Schwierigkeiten durch ausschließliche Verwendung von POKE-Anweisungen anstelle von DATA-Zeilen. Um den Aufwand so gering wie möglich zu halten, wurde angestrebt, mit möglichst weniger POKE-Anweisungen auszukommen. Das wurde mit einem Trick erreicht: Das für die Rückübersetzung der Token beim Ausdrucken nach dem LIST-Befehl zuständige Maschinenunterprogramm des Betriebssystems enthält fast alle Befehlssequenzen, die wir für unsere Zwecke benötigen. Das Programm in Listing 4 lädt den Maschinenprogrammabschnitt 50929 bis 51008 (C6F1 — C740), der zugegebenermaßen viel für unsere Zwecke überflüssigen Ballast enthält, mit einer einfachen FOR-NEXT-Schleife in den Kassettenpuffer, (Es wird mit GOSUB700 angesprungen). Jeder zweite und weitere Ansprung kann mit GOSUB730 erfolgen und wird dann in kaum wahrnehmbaren Bruchteilen einer Sekunde ausgeführt.

(Fred Behringer/ev)

```
     LDY #$FF        ZEICHENZAEHLER
NEXT INY             NAECHSTES ZEICHEN
     LDA $C09E,Y     IN AKKU
     BPL NORM        WORTENDE ?
     AND #$7F        DANN NORMALISIERT
     JSR $CB47       ZEICHEN AUSGEBEB
     LDA #$2E        WORTTRENNZEICHEN
NORM JSR $CB47       ZEICHEN AUSGEBEB
     BNE NEXT        ZEICHEN IN AKKU = 0 ?
ENDE RTS             DANN ZURUECK ZU BASIC
```

Listing 3. Assemblerdarstellung des nach Listing 2 erzeugten Maschinenprogramms

```
700 REM:HELP
710 FORI=0TO80:POKE828+I,PEEK(50929+I):N
EXT
720 POKE836,55:POKE892,169:POKE893,166:P
OKE897,6:POKE903,180:POKE909,96
730 POKE782,255:SYS898:RETURN
READY.
```

Listing 4. Simulation von HELP als Maschinenprogramm für Kassettenpuffer, ohne DATA-Zeilen. Länge 106 Bytes, Einlesen 1,5 s, Ausführung »augenblicklich«.

Genau betrachtet: RS232/V.24-Schnittstelle

Eine kurze und bündige Beschreibung der RS232-Schnittstelle Ihres C 64. Was machen die Signale, wie sind die Pin-Belegungen?

Bei der RS232-Schnittstelle werden die Daten Bit für Bit übertragen, im Gegensatz zur Centronics- oder IEEE-488-Norm, bei der ganze Bytes übergeben werden. Die Bits werden als eine Folge von Spannungsimpulsen mit einer bestimmten Dauer übertragen. In der Praxis werden dabei Pakete von 5 bis 8 Datenbit übertragen, die von einem Startbit und 1 bis 2 Stop-Bit eingerahmt sind (Bild 1). Das Startbit hat grundsätzlich logischen Low- und die Stop-Bits High-Pegel. Vor dem Stop-Bit kann ein sogenanntes Paritäts-Bit vereinbart werden, das die Anzahl der High-Zustände im Datenwort immer gerade oder ungerade macht.

Beispiel: Sind in einer 8-Bit-Übertragung 5 Bit gesetzt, wird das Paritäts-Bit ebenfalls gesetzt, wenn gerade Parität vereinbart wurde.

Päckchenweise Übertragung

Um die Störungs-Anfälligkeit der Übertragung zu mindern, wird logisch »Eins« (gesetztes Bit) nicht durch +5V (TTL-Pegel) realisiert, sondern mit einer Spannung von —3 bis —12V und logisch »Null« mit +3 bis +12V (RS232 nach DIN 66020). Eine andere Norm ist die RS232/TTY, die gegen äußere Störungen recht unempfindlich ist. Bei dieser Norm werden die logischen Zustände durch das Fließen oder Fehlen eines Stromes (20mA) dargestellt. Der C 64 hat zwar die nötige Software für eine RS232-Schnittstelle im Betriebssystem integriert, verfügt aber nicht über die entsprechenden Spannungspegel. Im C 64 gibt es nur zwei Spannungen: +5V (TTL) und 9V Wechselspannung. Es ist also ein Interface zur Spannungskonvertierung nötig. Links in Bild 2 finden Sie den Schaltplan eines solchen Interfaces (Bauanleitung in Ausgabe 3/85). Rechts im Bild die diskrete Lösung, für die Konvertierung von 0/5V auf ±12V (oben) und von ±12V auf 0/5V (unten). Beachten Sie, daß jede Sende- und Empfangsleitung die entsprechende Transistorschaltung braucht.

Mit einer Masse- und einer Datenleitung könnte schon eine Übertragung von Texten an einen Drucker erfolgen. Was ist aber, wenn die Datenübertragung schneller ist, als der

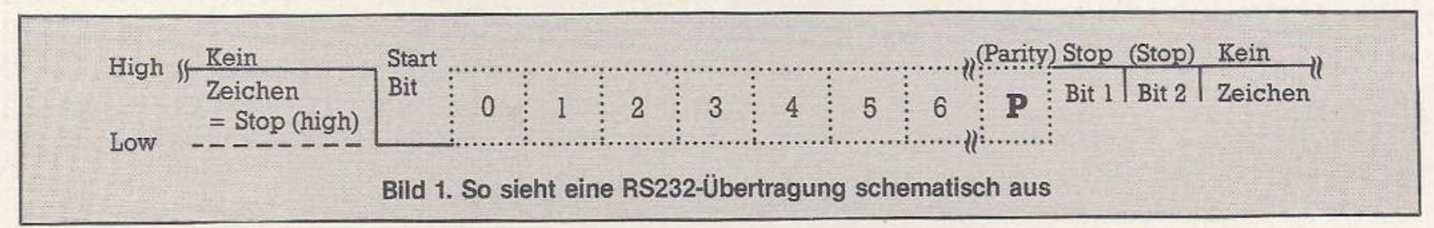


Bild 1. So sieht eine RS232-Übertragung schematisch aus

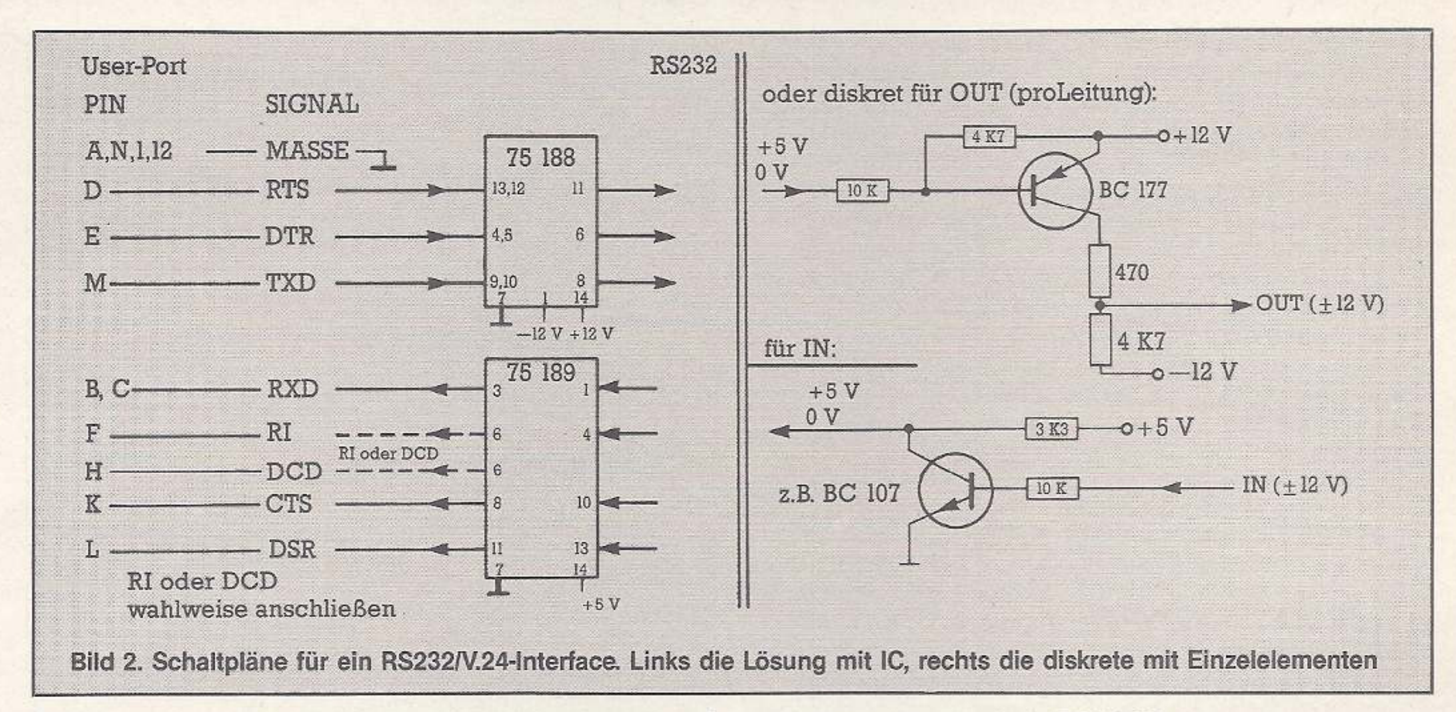


Bild 2. Schaltpläne für ein RS232/V.24-Interface. Links die Lösung mit IC, rechts die diskrete mit Einzelelementen

Drucker die Zeichen zu Papier bringen kann? Der Drucker muß dem Computer sagen, wenn er zuviel Arbeit bekommt. Er kann dies auf drei verschiedene Arten tun:

— Software-Protokoll mit XON/XOFF

Es wird eine zusätzliche Leitung zwischen Computer und Peripherie eingerichtet, über die das empfangende Gerät den Code $13 (XOFF) sendet, wenn es keine Daten mehr annehmen kann. Dieses Signal hat die gleiche Aufgabe wie die Busy-Leitung einer Centronics-Schnittstelle; es stoppt die Datenübertragung. Die Freigabe erfolgt mit dem Code $11 (XON). Die Codes $11 und $13 entsprechen den ASCII-Codes DC1 und DC2.

Die neue Leitung kann natürlich auch zur Übertragung von mehr Informationen verwendet werden. Sende- und Empfangsgerät müssen dann allerdings in der Lage sein, zwei Leitungen (XON/XOFF und die normale Datenleitung) gleichzeitig zu verwalten. Mit dieser zusätzlichen Leitung wird auch der sogenannte Vollduplex-Betrieb möglich. Vollduplex heißt, daß beide Geräte gleichzeitig senden oder empfangen können. Im Gegensatz zum Halbduplex-Betrieb, bei dem zur gleichen Zeit nur in eine Richtung übertragen werden kann.

— Software-Protokoll mit ETX/ACK

Auch bei dieser Lösung kommt man nicht ohne eine zusätzliche Leitung aus. Sie heißt DTR (Data Terminal Ready). Ist zum Beispiel der angeschlossene Drucker bereit, Daten anzunehmen, aktiviert er die DTR-Leitung und sendet $06 (ACKnowledge). Der Computer schickt nun eine Reihe Datenworte über die Sendeleitung und schließt die Übertragung mit $03 zwischendurch immer wieder ab. Den nächsten Datenblock sendet er erst dann, wenn der Drucker sein ACK gegeben hat. Damit die Übertragung nicht in einem Chaos entartet, muß der Sende-Computer über das Puffervermögen des Empfängers informiert sein, um rechtzeitig ein ETX (End Of Text) senden zu können. Nur so kann ein Überlauf des Puffers und der damit einhergehende Datenverlust verhindert werden.

Hat das Empfangsgerät ein ETX festgestellt, werden die empfangenen Daten bearbeitet. Kann der Empfänger neue Daten aufnehmen, sendet er ein ACK an den Computer und die Übertragung beginnt von Neuem.

— Hardware-Protokoll

Spätestens hier wird es unübersichtlich. Es hilft nur noch Tabelle 1 zur Erklärung der ganzen Signale. Als üblicher Stecker

Bit (dez. Wert)	Bedeutung
	1.) Kontrollregister (8 Bit)
Bit 7 (128)	0 = 1 Stop-Bit 1 = 2 Stop-Bits

Bit (dez. Wert)	8 Daten-Bits	7 Daten-Bits	6 Daten-Bits	5 Daten-Bits
Bit 6 (64)	0	0	1	1
Bit 5 (32)	0	1	0	1

Bit (dez. Wert)	Bedeutung
Bit 4 (16)	nicht benutzt

Baudraten (Bit/sec)

Bit (dez. Wert)	nicht impl.	50	75	110	134,5	150	300	600	1200	1800	2400
Bit 3 (8)	0	0	0	0	0	0	0	0	1	1	1
Bit 2 (4)	0	0	0	0	1	1	1	1	0	0	0
Bit 1 (2)	0	0	1	1	0	0	1	1	0	0	1
Bit 0 (1)	0	1	0	1	0	1	0	1	0	1	0

2.) Kommandoregister (8 Bit)

Bit (dez. Wert)				keine Paritätsüberprüfung (alle vier Komb.)	ung. Par.	ger. Par.	Bit 8:1 ohne Par.	Bit 8:0 ohne Par.
Bit 7 (128)	0	0	1	1	0	0	1	1
Bit 6 (64)	0	1	0	1	0	1	0	1
Bit 5 (32)	0	0	0	0	1	1	1	1

Bit (dez. Wert)	Bedeutung
Bit 4 (16)	0 Vollduplex 1 Halbduplex
Bit 3 (8)	nicht benutzt
Bit 2 (4)	nicht benutzt
Bit 1 (2)	nicht benutzt
Bit 0 (1)	0 Freilaufmodus (3-Draht) 1 Hardwarehandshake (X-Draht)

Tabelle 2. Funktion des Kontroll- und Kommandoregisters

für V.24-Signale hat sich ein 25-poliger D-Sub-Stecker (im Laborslang Cannon gennant) durchgesetzt. Die Bezeichnung der Kontakte ist gleich dreimal genormt: DIN 66020, CCITT V.24 (Comité Consultatif International Télégraphique et Téléphonique) und EIA RS232C (Electronic Industries Association). Die Bedeutung der Signale ist bei allen Normen gleich, nur die Signalpegel differieren. Die deutsche Norm verlangt, im Gegensatz zu den anderen, negative Logik.

Zum Anschluß einer RS232/V.24-Schnittstelle ist es in den allermeisten Fällen nicht nötig, alle Leitungen zu benutzen. So werden nur wenige unter Ihnen eine synchrone Datenübertragung mit zusätzlichem Clock-Signal realisieren, wie beim seriellen IEC-Bus des C 64. Normalerweise reichen die folgenden Leitungen aus:

1. eine Masseleitung
2. je Richtung eine Datenleitung
3. je Richtung eine Busyleitung

Punkt 1 und 2 dürften klar sein. Punkt 3 kann auf vielfältige Art realisiert werden. In aller Regel werden die Kontakte S2 (RTS) und M2 (CTS) benutzt. Die Erklärung erfolgt am besten an einem Beispiel: Der Drucker zeigt seine Empfangsbereitschaft an, indem er M2 aktiviert. Dieses Signal fragt der Computer ständig am Anschluß S2 ab. Ist M2 inaktiv, stoppt der Computer die Datenübertragung. Nehmen wir an, anstelle des Druckers sei ein Meßgerät angeschlossen, das nur ab und zu Anweisungen vom Computer bekommt und ansonsten sich um interne Aufgaben kümmert. Dann wäre es wenig vorteilhaft, wenn das Meßgerät ständig Befehle vom Computer erwartet; für die eigentlichen Meßaufgaben bliebe zu wenig Zeit. In diesem Fall gestattet das einfache Abfragen des Kontaktes S1 (per Interrupttechnik) eine fast ungestörte Bearbeitung eines Programmes. S1 ist mit RTS (Request To Send) identisch. Aktiviert der Computer S1, »spitzt das Meßgerät die Ohren« und das Meßprogramm verzweigt in die Datenempfangsroutine.

Nicht auf die Norm verlassen

Die Hersteller von V.24-Schnittstellen scheinen sich nicht immer völlig einig zu sein, wie die Belegung und Bedeutung der einzelnen Stecker-Pins ist. So sind diesem Beitrag hauptsächlich Praxiserfahrungen zugrunde gelegt. Besser als Normblätter ist die Überprüfung der Schnittstelle mit einem Speicheroszilloskop oder einem Digital-Analyzer. Zumal die Steuerleitungen ab und zu mit Fantasienamen belegt oder nicht eindeutig als Sende- oder Empfangsleitung gekennzeichnet werden. Zum störungsfreien Betrieb einer Schnittstelle sollten nichtbenutzte Leitungen auf ein festes Potential gelegt werden. Man verhindert dadurch, daß die Übertragung bei einer eventuellen Abfrage einer solchen Leitung, mit undefiniertem logischen Zustand, nicht unterbrochen wird.

V.24 beim C 64

Beim C 64 kann diese Schnittstelle als Gerät der Nummer 2 angesteuert werden. Zur Bestimmung der Kontrollparameter sind zwei Register vorhanden, die auch von Basic aus erreicht werden können.

Einstellparameter

Mit Tabelle 2 können Sie die Werte bestimmen, die Sie in Kommando- und Kontrollregister schreiben müssen, um ein bestimmtes Übertragungsprotokoll zu bewerkstelligen. Eine »1« bedeutet ein gesetztes Bit.

Das Einschalten der RS232-Schnittstelle geschieht beim C 64 mit OPEN filenr.,2,0,CHR$(Kontrollreg.)+CHR$(Kommandoreg.).

Beispiel: OPEN20,2,0,CHR$(64+4+2)+CHR$(32+1).

Mit dieser Anweisung wird für Filenummer 20 vereinbart: 1 Stop-Bit, 6 Datenbit, 300 Baud, ungerade Parität, Vollduplex und Hardwareprotokoll.

Pin/Bedeutung	DIN	CCITT	EIA	User-Port C 64 (VC 20)
1 Masse	E1	101	AA	A-GND (A-GND)
2 Transmit data (TD) Über diese Leitung sendet der C 64 Daten an den Akustikkoppler.	D1	103	BA	M-PA2 (M-CB2) out
3 Received data (RD) Die Empfangsleitung.	D2	104	BB	B-F12 + C-PB0 (B-CB1 + C-PB0)
4 Request to send (RTS) Frage des Computers an das Peripheriegerät, ob es zur Datenübertragung bereit ist.	S2	105	CA	D-PB1 (D-PB1) out
5 Clear to send (CTS) Positive Antwort des Peripherie-gerates auf RTS.	M2	106	CB	K-PB6 (K-PB6)
6 Data set ready (DSR) Akustikkoppler ist betriebsbereit	M1	107	CC	L-PB7 (L-PB7)
7 Signalmasse	E2	102	AB	N-GND (N-GND)
8 Received line signal (DCD)	M5	109	CF	H-PB4 (H-PB4)
9 Testzwecke				
10 Testzwecke				
11 nicht belegt				
12 Secondary carrier detector	HM5	122	SCF	—
13 Secondary clear to send	HM2	121	SCB	—
14 Secondary transmitted Data	HD1	118	SBA	—
15 Transmit clock (TC) from modem	T2	114	DB	—
16 Secondary received data	HD2	119	SBB	—
17 Receiver signal clock (RC)	T4	115	DD	—
18 nicht belegt				
19 Secondary request to send	HS2	120	SCA	—
20 Data terminal ready (DTR) Terminal ist zur Datenübertragung bereit.	S1.x	108.x	CD	E-PB2 (E-PB2) out
21 Signal quality detector	M6	110	CG	—
22 Ring indicator (RI)	M3	125	CE	F-PB3 (F-PB3)
23 Data signal rate det. terminal	S4	111	CH	—
modem	M4	112	CI	—
24 Transmit clock to modem	T1	113	DA	—
25 nicht belegt				

Tabelle 1. Belegung der RS232-Leitungen

Wird eine Filenummer größer als 128 verwendet, sendet der Computer (wie üblich) nach jedem Carriage Return ($0D) ein Linefeed ($0A). Sollten diese Einstellmöglichkeiten nicht ausreichen, können Sie direkt in die RS232-Routine eingreifen. Einige Speicherstellen finden Sie in Tabelle 3.

$0293	Kontrollregister
$0294	Kommandoregister
$0295—$0296	nicht-standard (Bit time 2/100)
$0297	Statusregister
$0298	Anzahl Bits
$0299—$029A	Baudrate
$029B	Zeiger Aufnahme
$029C	Zeiger Eingabe
$029D	Zeiger Senden
$029e	Zeiger Ausgabe

Tabelle 3. Zeropage-Adressen, die von der RS232-Schnittstelle benötigt werden.

Übertragungsraten größer als 2400 Baud, können nur mit eigenen Maschinenprogrammen realisiert werden.

Fehlerabfrage

Das Betriebssystem des C 64 verfügt über eine Fehlerbehandlung der RS232-Schnittstelle. Der Status dazu kann entweder durch die Variable ST abgefragt werden, oder direkt mit Speicherzelle $0297. Die Bedeutung der einzelnen Bits von ST finden Sie in Tabelle 4. (Jens Maßmann/hm)

Bit	dez.	Bedeutung
0	1	Paritätsfehler
1	2	Rahmenfehler
2	4	Empfängerpuffer voll
3	8	unbenutzt
4	16	CTS-Signal fehlt
5	32	unbenutzt
6	64	DSR-Signal fehlt
7	128	Break-Signal empfangen

Tabelle 4. Bedeutung der Statusvariablen

Longscreen VC 20

Diese kleine Utility wird sicher bei allen Anwendern Freude finden, die das unproportionale Breitschriftformat des VC 20 stört.

Die Routine dreht den kompletten Zeichen-, Farb- und Bildschirmspeicher um 90 °. Das Ergebnis ist eine hervorragend lesbare Bildschirmmatrix mit herrlich schlanken Zeichen (siehe Bild). Natürlich sind nach wie vor beide Zeichensätze vorhanden.

Für das Programm benötigt man mindestens eine 8-KByte-RAM-Erweiterung. Nach dem Start wird folgende Speicheraufteilung vorgenommen:

Bereich	Grenze
Basic-RAM	
	$1d18 (7448)
Maschinenprogramm	
	$1c00 (7168)
Grafik-RAM (2K)	
	$1400 (5120)
Video-RAM	
	$1200 (4608)
Print-RAM	
	$1000 (4096)

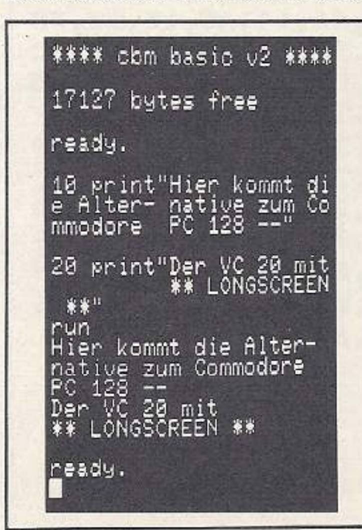
Die Position des Speichers für Print-Operationen bleibt also. Falls in diesen Bereich gePOKEt wird, werden solche Zeichen auch übertragen. Der VIC zeigt jetzt allerdings nicht mehr das Print-RAM, sondern das Video-RAM, wo der Inhalt des Print-RAM um 90 Grad gedreht dargestellt wird. Die Routine wird über das IRQ-Signal gesteuert. Das Bild wird 12mal pro Sekunde aufgebaut.

Folgende RAM-Adressen werden benutzt:

$00/01	Zeiger in Video-RAM
$02/03	Zeiger in Print-RAM
$06/07	Zeiger in Video-Farb-RAM
$08/09	Zeiger in Print-Farb-RAM
$04	Y-Hilfsregister für Print-RAM
$05	Y-Zeiger in Video-RAM

Sollte die RESTORE-Taste betätigt worden sein, so befindet sich der Rechner wieder im Normalmodus. Durch Eingabe des Befehls »SYS 7410« kann die Routine ohne Programm-Verlust wieder gestartet werden. Die Umschaltung in den Grafik-/Großschrift-Modus erfolgt über »PRINT CHR$ (142)«, in den Textmodus zurück gelangt man über »PRINT CHR$ (14)«.

(Wolfgang W. Wirth/ev)



Den Monitor auf die Seite gelegt, und schon hat man eine hervorragend lesbare Bildschirmanzeige

```
100 REM--------------                      <105>
105 REM! LONGSCREEN !                      <042>
110 REM! VERSION 2  !                      <151>
115 REM--------------                      <120>
120 REM                                    <007>
125 REM--------------                      <130>
130 REM!    BY      !                      <238>
135 REM! W.WIRTH    !                      <107>
140 REM--------------                      <145>
145 REM! 06703/868  !                      <055>
150 REM--------------                      <155>
155 REM                                    <042>
160 POKE 56,27:CLR                         <013>
165 POKE 36879,12                          <220>
170 PRINT"{CLR,DOWN,RIGHT,WHITE,CTRL-N}LAD
    EPRG. FUER :"                          <244>
175 PRINT"{DOWN,RIGHT,RVSON}LONGSCREEN VER
    SION 2{3DOWN}"                         <071>
180 FOR I=7168 TO 7489                     <134>
185 READ J:POKE I,J:S=S+J                  <160>
190 PRINT I;J"{UP}":NEXT                   <182>
195 IF S=37513 THEN 210                    <144>
200 PRINT"{2DOWN,RIGHT}CHECKSUMMENFEHLER !
    "                                      <238>
205 PRINT"{DOWN,RIGHT}DATA-ZEILE PRUEFEN !
    {DOWN}":STOP                           <169>
210 PRINT"{RIGHT}RESTORE-TASTE NOCH"       <198>
215 PRINT"{DOWN,RIGHT}INTAKT!"             <206>
220 PRINT"{DOWN,RIGHT}FALLS BETAETIGT, MIT
    "                                      <008>
225 PRINT"{DOWN,RIGHT,RVSON}SYS 7410{RVOFF
    ,SPACE}NEU STARTEN"                    <100>
230 PRINT"{2DOWN,RIGHT}ZUM START BITTE"    <004>
235 PRINT"{DOWN,RIGHT}F1-TASTE DRUECKEN"   <108>
240 POKE 198,0:WAIT 198,1:SYS 7448         <255>
245 DATA 206,255,017,016,106,169,004,141   <112>
250 DATA 255,017,162,010,181,255,072,202   <114>
255 DATA 208,250,160,016,134,002,132,003   <104>
260 DATA 160,148,134,008,132,009,160,022   <122>
265 DATA 132,005,056,162,227,160,019,134   <132>
270 DATA 000,132,001,160,151,134,006,132   <111>
275 DATA 007,160,021,177,002,170,177,008   <140>
280 DATA 132,004,164,005,145,006,138,145   <144>
285 DATA 000,164,004,165,000,233,023,133   <132>
290 DATA 000,133,006,176,005,198,001,198   <159>
295 DATA 007,056,136,016,222,165,002,233   <158>
300 DATA 234,133,002,133,008,144,004,230   <151>
305 DATA 003,230,009,198,005,016,187,162   <177>
310 DATA 246,104,149,010,232,208,250,076   <179>
315 DATA 191,234,072,165,154,201,003,240   <178>
320 DATA 003,076,133,242,104,072,133,215   <180>
325 DATA 138,072,152,072,165,212,240,003   <189>
330 DATA 076,073,231,165,215,201,014,240   <193>
335 DATA 007,201,142,208,243,162,128,172   <201>
340 DATA 162,136,160,000,132,096,134,097   <211>
345 DATA 162,020,132,098,134,099,162,008   <222>
350 DATA 177,096,145,098,200,208,249,230   <239>
355 DATA 097,230,099,202,208,242,160,008   <231>
360 DATA 132,099,160,020,134,096,132,097   <238>
365 DATA 162,007,160,007,169,000,133,098   <235>
370 DATA 177,096,010,145,096,038,098,136   <008>
375 DATA 016,246,165,098,072,202,016,234   <252>
380 DATA 160,007,104,145,096,136,016,250   <248>
385 DATA 165,096,024,105,008,133,096,144   <007>
390 DATA 215,230,097,198,099,208,209,076   <033>
395 DATA 220,230,120,162,000,160,028,142   <240>
400 DATA 020,003,140,021,003,162,114,142   <238>
405 DATA 038,003,140,039,003,088,162,005   <013>
410 DATA 189,018,029,157,000,144,202,016   <025>
415 DATA 247,096,012,038,151,044,097,205   <039>
420 DATA 120,032,141,253,032,082,253,032   <018>
425 DATA 249,253,032,024,229,032,091,228   <045>
430 DATA 162,024,160,029,032,138,254,032   <040>
435 DATA 242,028,169,014,032,210,255,141   <044>
440 DATA 015,144,169,001,141,134,002,076   <045>
445 DATA 123,227                           <158>
```

Listing »Longscreen« für den VC 20.

C 16: HELP und TRACE verbessert

Dieses kleine Programm implementiert eine wesentlich erweiterte TRACE-Funktion. Auch HELP wird damit um einiges übersichtlicher.

Nachteilig bei der bisherigen Fehlersuche durch »HELP« ist das entnervende Blinken der Fehlerstelle und aller nachfolgenden Zeichen. Eine konzentrierte Fehlersuche ist kaum möglich. Durch die Änderung dieser Routine wird die Fehlerstelle fortan nicht mehr blinkend, sondern in reverser Schreibweise ausgegeben und auch nur diese einzige Stelle, nicht mehr die gesamte Restzeile. Der Fehler ist somit mit einem einzigen Blick zu erfassen.

Das C16-Tracing mag etwas für Leute mit Facettenaugen sein, aber wohl nichts für die Mehrzahl der Anwender. In der durch dieses Programm erzeugten neuen Version werden nicht mehr wild die abgearbeiteten Zeilennummern ausgegeben, sondern die gesamte aktuelle Programmzeile gezeigt. Der Befehl, auf dem der Programmzeiger gerade steht, erscheint revers geschrieben. Nach jeder Zeilenausgabe stoppt das abzuarbeitende Programm, und die Trace-Routine wartet auf irgendeinen Tastendruck.

Danach wird die angezeigte Zeile bis zum folgenden Trennzeichen ausgeführt. Dieser Vorgang wiederholt sich bis zum Programmende. Der Programmablauf läßt sich wie gewohnt mit der STOP-Taste unterbrechen.

Vor allem für Basic-Anfänger ist dieses verbesserte Trace eine wertvolle Hilfe, da sich das Programm bei der Ausführung von Befehl zu Befehl direkt verfolgen läßt.

(Wolfgang W. Wirth/ev)

```
100 REM ***************************
110 REM *                         *
120 REM * EXTENDED HELP & TRACE   *
122 REM *                         *
124 REM *      C 16 / C 116       *
130 REM *                         *
134 REM *                         *
140 REM * BY WOLFGANG WIRTH       *
144 REM *                         *
150 REM *   THEODOR-HEUSS-RING    *
154 REM *                         *
160 REM *   6556 WOELLSTEIN       *
170 REM *                         *
180 REM ***************************
190 REM
200 REM
220 ADRESSE=819:ANZAHL=8
230 FOR ZEILE=410 TO 760 STEP 10
240 SUMME=0
250 IF ZEILE=760 THEN ANZAHL=3
260 FOR SPALTE=1 TO ANZAHL
270 READ BYTE$:BYTE=DEC(BYTE$)
280 SUMME=SUMME+BYTE AND 255
290 POKE ADRESSE,BYTE
300 ADRESSE=ADRESSE+1
310 NEXT
320 READ TEST$
330 IF SUMME=DEC(TEST$) THEN 360
340 PRINT"FEHLER IN ZEILE";ZEILE
350 FLAG=1
360 NEXT
370 IF FLAG THEN END
380 SYS 1082
400 REM
410 DATA38,66,53,A0,03,84,49,84,E5
420 DATA0F,20,5F,A4,A9,20,A4,49,E8
430 DATA29,7F,20,B2,90,C9,22,D0,C5
440 DATA06,A5,0F,49,FF,85,0F,C8,5E
450 DATAF0,09,A2,00,86,C2,24,53,5A
460 DATA10,19,A6,60,98,18,65,5F,A3
470 DATA90,01,E8,EC,F6,04,D0,0B,3A
480 DATACD,F5,04,90,06,F0,04,26,76
490 DATAC2,46,53,20,D1,04,F0,E3,23
500 DATA10,C8,C9,FF,F0,C4,24,0F,87
510 DATA30,C0,AA,84,49,A0,81,84,0C
520 DATA23,A0,8E,84,22,A0,00,CA,61
530 DATA10,0F,B1,22,48,E6,22,D0,12
540 DATA02,E6,23,68,10,F4,30,EF,96
550 DATAC8,B1,22,30,99,20,B2,90,C6
560 DATAD0,F6,60,AE,EF,04,E8,F0,9F
570 DATA19,AD,F0,04,AC,F1,04,85,E0
580 DATA14,84,15,20,3D,8A,90,0A,2E
590 DATA20,3E,90,A6,14,A5,15,20,82
600 DATA33,03,4C,3E,90,20,73,04,E7
610 DATA20,D9,03,4C,DC,8B,F0,83,22
620 DATA2C,EB,02,10,2E,24,81,10,0C
630 DATA2A,48,A4,3C,A6,3B,D0,01,04
640 DATA88,CA,8E,F5,04,8C,F6,04,5F
650 DATAA5,39,A4,3A,85,14,84,15,EE
660 DATA20,3D,8A,A6,14,A5,15,20,7B
670 DATA33,03,20,3E,90,20,DD,EB,0C
680 DATAF0,FB,68,C9,EA,F0,03,4C,45
690 DATA3F,8C,20,73,04,4C,AE,03,5F
700 DATAA2,FF,86,3A,20,5A,88,86,E9
710 DATA3B,84,3C,20,73,04,AA,F0,2C
720 DATAEF,90,09,20,53,89,20,79,1D
730 DATA04,4C,D3,03,4C,2E,87,A2,C9
740 DATA03,8E,09,03,E8,8E,03,03,19
750 DATAA2,D0,8E,08,03,A2,1B,8E,56
760 DATA02,03,60,65

READY.
```

Listing »Extended Help & Trace« für den C 16


64ER ONLINE

Basic-Wegweiser für den Commodore 64

Hinter diesem eher bescheiden wirkenden Titel verbirgt sich mehr als ein bloßes Remake des C 64-Handbuchs, wie es zur Zeit leider allzu oft in der nahezu unüberschaubaren Fachbuchlandschaft anzutreffen ist. Vielmehr bietet dieses im Rahmen der »Wegweiser«-Reihe des Wiesbadener Vieweg-Verlags erschienene Buch auf 244 Seiten einen interessanten Einstieg in das weite Feld der Datenverarbeitung, der nicht nur dem Laien einiges Neue vermitteln mag.

Beginnend mit grundlegenden Begriffsklärungen wie »Was ist Hardware, Software, Firmware?« oder »Welche Unterschiede gibt es zwischen Großrechnern und Mikrocomputern?«, erläutert Autor Dr. Ekkehard Kaier dem Leser ausführlich die Funktion von Betriebssystem und Anwenderprogramm und stellt kurz die derzeit wichtigsten Programmiersprachen vor. Auch auf die fundamentalen Techniken des Anlegens von Programm- und Datenstrukturen wird in diesem ersten von drei Abschnitten des Buches eingegangen. Erst nachdem der Benutzer seinen Computer in den Gesamtrahmen der Informatik einzuordnen vermag, wird er von Expertenhand behutsam in die Welt seines C 64 geführt.

Dies macht sich der zweite Teil der Lektüre zur Aufgabe, an dessen Anfang eine gründliche Einweisung in Tastatur, Bildschirm und Diskettengerät steht. Anschließend lernt der Leser anhand kleiner Beispielroutinen den gesamten Basic-Wortschatz seines C 64 kennen und erstellt sein erstes Programm. In diesem Zusammenhang wird auch kurz auf die Spracherweiterungen Basic 4.0 und Simons Basic, sowie auf die Kompatibilität von C 64-Programmen zu Computern der anderen Commodore-Serien hingewiesen. Nach dem Studium dieses zweiten Abschnitts sollte der Benutzer seinen C 64 selbst bedienen und einfache Programme erstellen können.

Im dritten und mit 128 Seiten weitaus umfangreichsten Teil des »Basic-Wegweisers« erfolgt eine gründliche Einarbeitung in die Basic-Programmierung des C 64. Der Einsatz von Folge-, Auswahl-, Wiederholungs- und Unterprogrammstrukturen, das Suchen, Sortieren und Mischen von Daten lehrt den Leser systematisch zu programmieren. Probleme der Stringverarbeitung werden hier erörtert, Wege zur Behandlung sequentieller und Direktzugriffsdateien aufgezeigt. Alle theoretisch erarbeiteten Inhalte erhalten durch aussagefähige Beispielprogramme praktischen Bezug. Bei einem abschließenden Ausflug in Simons Basic soll sich der Benutzer mit Musikprogrammierung, hochauflösender- und Spritegrafik vertraut machen. Unglücklicherweise ist dieses Kapitel für Nichtbesitzer von Simons Basic — vermutlich die Mehrzahl der Leser gänzlich nutzlos.

Von diesem geringfügigen Mangel abgesehen, präsentiert sich mit »Basic-Wegweiser für den Commodore 64« ein außergewöhnlich klar strukturiertes und inhaltsstarkes Buch, von dem selbst bei einem Preis von 38 Mark nicht nur der Verleger profitiert.

Die im Buch abgedruckten Lehrprogramme können gegen 42 Mark bei Vieweg auf Diskette angefordert werden. (Jörg Veit)

Dr. E. Kaier, Basic-Wegweiser für den Commodore 64, Vieweg & Sohn Verlag GmbH, 244 Seiten, ISBN 3-528-04303-2, 38 Mark.

Dienstprogramme VC 20, Commodore 64 und Executive

Im Verlauf der letzten 12 Monate hat es sich herumgesprochen, daß Computerliteratur ein Renner auf dem Büchermarkt ist. Das hat zu allerlei merkwürdigen »Buchblüten« geführt: Papier ist geduldig! Der durch Schulbücher und wissenschaftliche Werke renommierte Vieweg-Verlag hat die Zeichen der Zeit erkannt. Auf einen Vertrauensvorschuß beim Käufer bauend, bietet er inzwischen den 8. Band seiner Reihe »Anwendung von Mikrocomputern« an, wobei er großzügig auch Homecomputer wie den VC 20 oder den Commodore 64 einschließt.

Ernst-Friedrich Reinking wendet sich mit diesem Buch sowohl an den »weniger vorgebildeten« als auch an den »guten« Programmierer. Tatsächlich bietet er auch jedem etwas:

Ein Drittel des Inhaltes führt in die Assembler-Programmierung ein. Wer allerdings erwartet, hier auf knapp 32 Seiten wirklich Assembler-Programmierung lernen zu können, muß enttäuscht werden: Mehr als eine kleine Gedächtnisstütze ist auf so wenig Platz nicht unterzubringen. Ganz nett ist die Auflistung der Kernal-Routinen, die jeweils noch mit einem kleinen Beispiel gewürzt sind.

Danach geht's erst richtig los: Wem noch Utilities wie AUTONUMBER, RENUMBER (mit GOTO, GOSUB, ...), MERGE, TRACE oder SINGLE-STEP in seiner Programmsammlung fehlen, der findet diese Ergänzungen hier. Und das in Maschinensprache, sauber programmiert und sehr gut erklärt. Außerdem findet man noch eine DUMP-Funktion (für die einfachen Variablen), ein Programm zum Hervorheben von REM-Zeilen und SEARCH, was erlaubt, aus einem Basic-Programm beliebige Suchbegriffe mit Angabe der Zeilennummer herauszufinden. Die Eingabe des Gesuchten ist allerdings etwas eigentümlich: Eine Zeile 0 mit dem Suchbegriff muß vor das Basic-Programm gehängt werden.

Kritik finden muß auch eine ziemlich unsinnige Routine, die sich etwas hochtrabend »+/-Scrolling« nennt, aber nichts weiter tut, als ein Programm auf reichlich unbequeme Weise Zeile für Zeile zu listen.

Und weil gerade das Negative dran ist: Warum kann Reinking die an sich ganz nützlichen Sortierroutinen HEAPSORT nicht auch in Maschinensprache anbieten? In Basic sind sie wirklich zu langsam! Außerdem ist es ein Jammer, daß er gerade hier von der Praxis alles sehr deutlich zu erklären abweicht und den Leser mit einem etwas undurchschaubaren Flußdiagramm abspeist. Durch ein Programm »Sortiertes Directory« (in Basic) und ein weiteres, »Unscratch«, welches — zwar auch langsam, weil in Basic — versehentlich gelöschte Files auf der Diskette wieder herstellt, wird man allerdings ganz gut entschädigt. Ein »UNNEW«, durch das mittels NEW oder Reset gelöschte Basic-Programm vom Interpreter wiedergefunden werden, eine Hardcopy-Routine, die den Inhalt des normalen Bildschirmes per Drucker für die Nachwelt fixiert (allerdings mit anderem Zeilenabstand, was sich bei Grafiken nachteilig auswirkt) fehlen ebensowenig wie ein kleiner Disassembler (ebenfalls in Basic).

Alle Programme sind sowohl für den VC 20, als auch den C 64 und den Executive (SX 64) ausgelegt. Lediglich zwei wurden speziell auf den VC 20 zugeschnitten: Ein Hardcopy-Programm, das einen selbstdefinierten Zeichensatz mitberücksichtigt und ein Programm zum Zeichnen von Funktionen auf dem Bildschirm, was alle VC 20-Besitzer ohne Supererweiterung freuen wird. Im Anhang findet sich noch eine Liste von nützlichen Interpreter-Routinen. Leider kann man aber ohne ROM-Listing nicht viel damit anfangen: Es fehlen alle Angaben darüber, wie man Parameter übergibt oder Ergebnisse abruft.

Resümee: Ein trotz der geschilderten Kritik empfehlenswertes Buch für den fortgeschrittenen Anfänger, der aus den vorgestellten Assembler-Routinen viel lernen kann.

(Heimo Ponnath)

Info: Ernst Friedrich Reinking, Dienstprogramme VC 20, Commodore 64 und Executive, Vieweg & Sohn 1984, ISBN 3-528-04299-0, 26,80 Mark.

Grafik auf dem Commodore 64

Geht man in eine Buchhandlung und blickt in die Sammlung von Büchern über den Commodore 64, so sieht man auf den Titeln immer wieder zwei Wörter: Basic und Grafik.

Mit letzterem beschäftigt sich eine Veröffentlichung des durch seine Schulbücher bekannten Westermann-Verlags.

In »Grafik auf dem C 64« wird allerdings nicht, wie schon öfters geschehen, erklärt, wie man selbige mit viel POKEs und Tricks auf die heimische Mattscheibe bekommt. Vielmehr beschäftigt es sich mit den vielfältigen Anwendungsmöglichkeiten der Grafikbefehle von Simons Basic.

Und so wimmelt es denn auf den vorliegenden 210 Seiten nur so von Programmbeispielen. Dabei sind unter anderem ein hervorragender Funktionsplotter, der allerdings noch einiges mehr kann, oder ein Programm zum Zeichnen dreidimensionaler Funktionen ohne die sogenannten »verdeckten Linien«.

Doch bei alledem kommt auch die Theorie nicht zu kurz. So erfährt man beispielsweise wie man Kurven, also nicht einfach Kreise oder Ellipsen, sondern beliebig gekrümmte verschlungene Gebilde nach dem Bezier-Verfahren zeichnen kann, oder wie man ein Niveaulinienbild einer dreidimensionalen Funktion erhält. Manchmal wird allerdings über das Ziel hinausgeschossen, wenn es um Differentialgleichungen oder gar Differentialgleichungssysteme geht. Hier ist zum genauen Verständnis schon ein gehöriger Schuß Oberstufenmathematik notwendig. Insgesamt gesehen ist das Buch doch recht mathematisch gehalten, aber immer noch ganz gut lesbar.

Zu erwähnen wäre vielleicht noch, daß natürlich alle Programmbeispiele erklärt werden. Sogar der Befehlssatz von Simons Basic wird, sofern er die Grafik betrifft, verständlich gemacht.

Ein klares Fazit: Empfehlenswert für jeden, der Grafik nicht nur verstehen, sondern auch ausnutzen möchte.

(Boris Schneider)

Walter Bachmann, Grafik auf dem C 64, Westermann-Verlag 1984, 204 Seiten, ISBN 3-14-508811-4, 39 Mark.


64ER ONLINE


64ER ONLINE


64ER ONLINE

Einige POKEs für den VC 20

Im folgenden ist X immer eine Zahl zwischen 0 und 255.

POKE 36865,X: Zentriert den Bildschirm in vertikaler Richtung. Man kann dadurch den Bildschirm nach oben oder unten verschieben. Der Normalzustand wird mit X=38 erreicht.

POKE 36864,X: Dieser Befehl ist für die horizontale Bildzentrierung zuständig. Er verschiebt den Bildschirm nach links oder rechts. Der Normalwert ist X=12.

POKE 37879,X: Mit diesem Befehl wird die interne Uhr des VC 20 beeinflußt. Man kann sie schneller oder langsamer laufen lassen. Die letzte Möglichkeit ist besonders beim LISTen interessant. Drückt man nämlich bei verlangsamtem Zeitgeber zusätzlich noch die CTRL-Taste, dann kann man sich einzelne Zeilen fast beliebig lange betrachten. POKE 37879,72 stellt den Normalzustand wieder her. (Detlef Krischak)

Basic-Programme retten

Die Betriebssystemroutine »Angleich von Koppeladressen« ab Adresse 42291 ermöglicht ein schnelles und einfaches »UNNEW« nach einem versehentlichen »NEW« oder Reset:
POKE 2049,1 : POKE 2050,1 : SYS 42291
Danach kann zumindest wieder geLISTet werden. Ein vollständiges »UNNEW« verlangt allerdings die Korrektur der Zeiger auf den Beginn der Variablen und Felder. Dazu wäre die Kenntnis der Programmlänge notwendig. Man kann sich aber behelfen, indem man das Programm notfalls in Teilen auf dem Bildschirm auf LISTet und die einzelnen Zeilen mit der RETURN-Taste neu übernimmt. (Gerhard Wagner)

Spezialeffekt

Wenn man beim C 64 in die Speicherstelle 53270 Werte zwischen 0 und 15 schreibt (POKE 53270,x), kann man den Bildschirm um bis zu sieben Bildpunkte nach links oder rechts scrollen lassen. Ist x kleiner als 8, dann scrollt der Bildschirmausschnitt um x Bildpunkte nach links, sonst um x-8 Bildpunkte nach rechts.

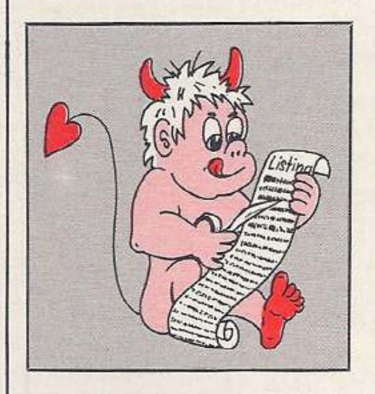
POKE 53270,8 stellt den Normalzustand wieder her.

Dieser Trick läßt sich gut bei Action-Spielen als optische Untermalung beispielsweise einer Explosion einsetzen. (Michael Keukert)

In C 64-Spielen gePOKEt

Hier sind einige interessante POKE-Befehle, mit denen man jeden Highscore überbieten kann. Doch Vorsicht, diese Befehle funktionieren nicht bei allen Versionen dieser Spiele.

* Fort Apocalypse: »POKE 14697,0 : POKE 14760,0 : POKE 36366,0«. Danach hat man beliebig viele Hubschrauber, einen unendlichen Treibstoffvorrat, und der Bonus wird nie erniedrigt.
* Hunchback: »POKE 9521,234 : POKE 9522,234 : POKE 9523,234«. Hier hat man unendlich viele Helden zur Verfügung.
* Neptune: »POKE 7870,60«. Mit diesem POKE hat man auf einen Schlag 60 Taucher.
* Jungle Hunt: »POKE 2242,234 : POKE 2243,234«. Der Held hat unendlich viele Leben. (Frank Bastian)



Fehlerteufelchen

Nachdem es nach Erscheinen der Ausgabe 4/85 des 64'er Magazins einige Tage verdächtig ruhig blieb (hatte das Fehlerteufelchen Urlaub genommen?), zerrann die Hoffnung auf eine fehlerfreie Ausgabe dann doch noch. Hier die Korrekturen:

Epson bedruckt Ostereier, Ausgabe 4/85, Seite 50

Die meisten Leser haben erkannt, daß es sich bei diesem Artikel um unseren Beitrag zum 1. April handelte. Alle Anfragen bezüglich des Bausatzes oder Abänderungen für Orangen oder Kieselsteine sind daher leider negativ zu beantworten.

11 neue Einzeiler, Ausgabe 4/85, Seite 153

Beim Einzeiler »Zeilen löschen am Bildschirm« muß es in Zeile 50 POKE 781,ZN heißen.

Der Einzeiler »Zugriffszeit der Floppy verkürzen« bringt wegen des Funktionsprinzips bei sequentiellen Dateien natürlich keinen Geschwindigkeitsgewinn.

Basic-Programme auf Trab gebracht — Compiler im Test, Ausgabe 2/85, Seite 38

Die vollständige Adresse, unter der der Petspeed bestellt werden kann, lautet:
Infotronik, Dipl.-Ing. Rolf Dahlen, Birkenstr. 40, 4100 Duisburg 17.

Durch die fehlende Ortskennzahl waren einige Briefe etwas länger unterwegs, als dies normalerweise üblich ist.

xBasic 64, Ausgabe 4/85, Seite 52

Aufgrund einer Vertauschung paßt die abgedruckte Befehlserklärung in einigen Punkten nicht zu dem Programm. Folgende Änderungen ergeben sich dadurch:
Beim Befehl HRG entfallen die Parameter x und y.
Der Befehl »TEXT« muß heißen »NRM«. Der Befehl »NEGATE« muß heißen »INVERS«. Der Befehl »INVERS« wird ersetzt durch »RESET«.
Syntax: RESET (x,y). Es gilt das unter SET gesagte, jedoch wird der Punkt gelöscht.
Der Befehl »AT« muß heißen: @ PRINT.
Syntax: @ PRINT s,z, "Text" oder @ PRINT s,z,Variable.

Hinzu kommt der Befehl ROM. Effekt: Nach Ausführung dieses Befehls holt sich der Computer die Bitmuster für die Zeichendarstellung wieder aus dem ROM.

Die Befehle KILL, DELETE, AUTO, DOKE aus Tabelle 1 haben in der abgedruckten Version keine Bedeutung.

SMON — Teil 5, Ausgabe 4/85, Seite 64

Punkt 4, Seite 67, ist leider falsch. Es muß beim Verschieben der »LDY #CF« in Adresse $9f71 in »LDY #9F« geändert werden.

Unter Punkt 5 ist der Bereich falsch angegeben. Es muß natürlich »M 9FD8 9FE4« eingegeben werden, sonst wird der alte und nicht der neue SMON geändert. Unangenehmer ist ein Programmfehler: Beim Einlesen eines Blocks wird das letzte Byte nicht (!) in den Speicher übernommen, wohl aber beim Zurückschreiben ein zufälliger Wert als 256. Byte. Abhilfe:

Zwei Befehle müssen mit »NOP« überschrieben werden. Zuerst mit SMON »O CED8 CEDC EA« eingeben, dann das Programm mit »S " :SMON $C000 D000« wieder abspeichern.

Text gut im Griff, Ausgabe 4/85, Seite 38

Die einzelnen Bewertungen des Textverarbeitungsprogramms »Protext 64« von S+S Soft bezogen sich auf Angaben des Herstellers. Die mit »Ja« beantworteten Kriterien sind deshalb mit Vorsicht zu genießen. Es entsteht der Eindruck, daß dieses 9,80 Mark-Programm mit wesentlich teureren konkurrieren könnte. Ein Test in einer der nächsten Ausgaben wird dies klarstellen.

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »64er« bietet allen Computerfans die Gelegenheit, für nur 5,— DM eine private Kleinanzeige mit bis zu 5 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **Juni-Ausgabe** (erscheint am 17. Mai 85): Schicken Sie Ihren Anzeigentext bis zum 19. April 85 (Eingangsdatum beim Verlag) an »64er«. Später eingehende Aufträge werden in der **Juli-Ausgabe** (erscheint am 14. Juni 85) veröffentlicht.

Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes.
Bitte beachten Sie: Ihr Anzeigentext darf maximal 5 Zeilen mit je 32 Buchstaben betragen.
Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, 64er« oder schicken Sie uns DM 5,— als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 11,— je Zeile Text veröffentlicht.

Private Kleinanzeigen

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen




64ER ONLINE


64ER ONLINE

Datum/Unterschrift


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE

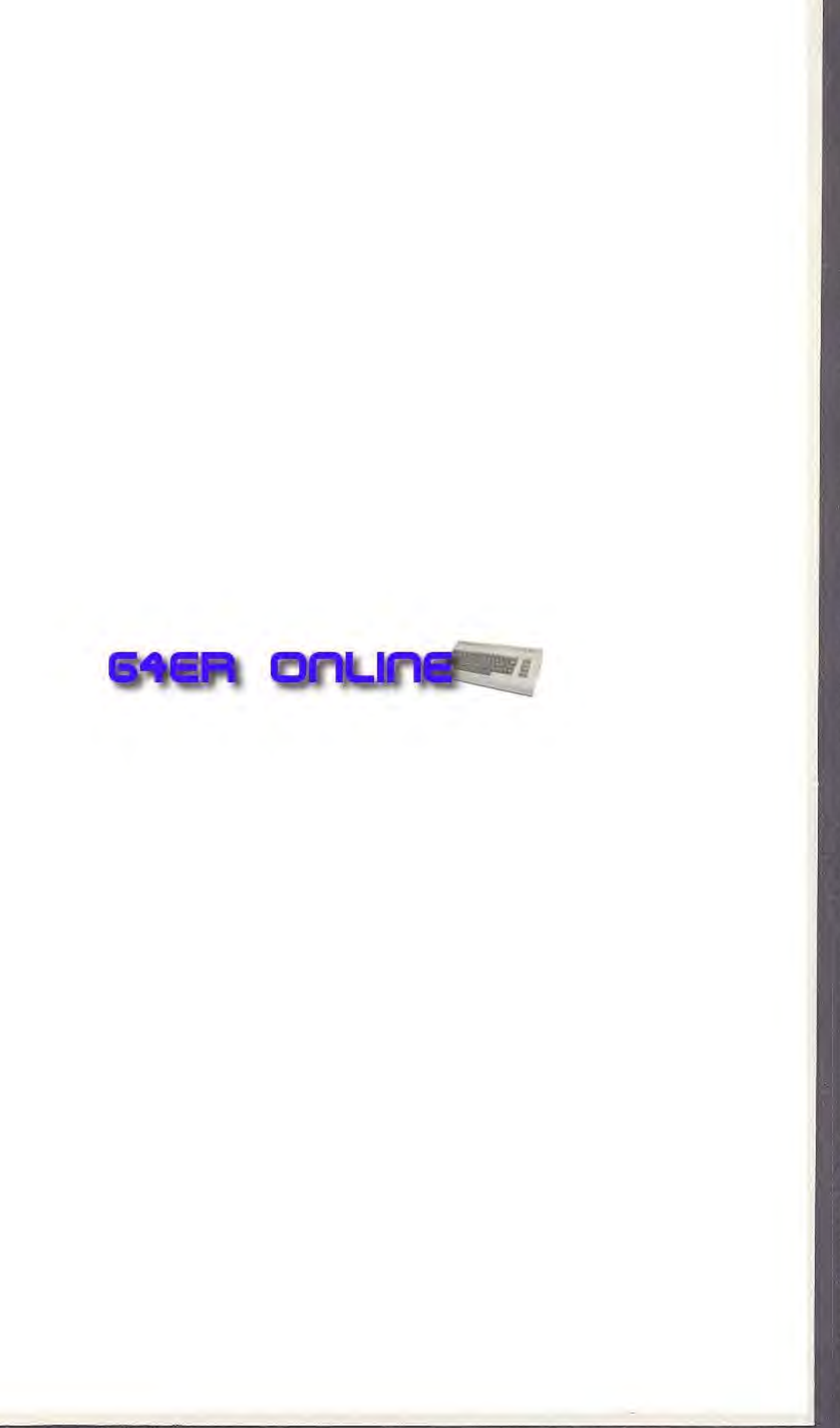

64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE


64ER ONLINE

Gewerbliche Kleinanzeigen



Gewerbliche Kleinanzeigen




64ER ONLINE


64ER ONLINE

Gewerbliche Kleinanzeigen

Gewerbliche Kleinanzeigen




64ER ONLINE

Der Ada-Trainingskurs auf dem C 64

Ada wurde im Jahre 1975 vom amerikanischen Verteidigungsministerium in Auftrag gegeben, um die Schwierigkeiten und die hohen Kosten in den Griff zu bekommen, welche durch die Vielfalt konkurrierender Programmmiersprachen entstanden sind. Jetzt gibt es auch auf dem C 64 die Möglichkeit, mit dieser modernen Sprache zu arbeiten.

Nur einige wenige Programmiersprachen — sogenannte Hochsprachen — haben eine weite Verbreitung gefunden. Cobol in der Wirtschaft, Fortran und Pascal in der Naturwissenschaft und Basic in der Home-Computerei.

Im riesigen Anwendungsgebiet der Industrie und Technik aber sind Hochsprachen selten zu finden.

Warum? Nun, die meisten für Steuerungen eingesetzten Computer sind nur mit einer bestimmten Hochsprache ausgestattet, die aber spezielle Eigenschaften des Computers oft nicht oder nur umständlich ausnutzt. Eine solche Hochsprache ist daher für diese Zwecke in der Regel nicht genügend effizient, zu langsam und zu aufwendig im Speicherbedarf.

Industrie-Programmierer greifen deshalb viel lieber auf Assemblersprachen zurück, mit dem Ergebnis, daß Programme nicht auf andere Computer übertragbar sind, meistens nur vom Autor selber verstanden und verbessert und nicht zu größeren Programmblöcken zusammengefügt werden können.

Selbst dann, wenn Programmierer die gleiche Hochsprache benutzen, müssen sie oft den Umgang mit neuen Betriebssystemen und anderen Programmentwicklungsmitteln lernen, beispielsweise Text-Editoren, Grafiksysteme, Konfigurationskontrollen und Fehlersuchhilfen. Schließlich sind Programme, die es gestatten, von einem System in ein anderes zu übersetzen, sehr aufwendig und teuer, genauso wie das Um- und Dazulernen der Programmierer.

Kein Wunder also, daß ein so großer Auftraggeber und Selbstverbraucher von Software wie das Verteidigungsministerium in den USA eine Hochsprache forderte, die alle diese Mängel abstellen sollte. Ganz nebenbei sollte diese Sprache stark strukturiert sein, eine einfache Fehlerbehandlung ermöglichen, mit unterschiedlicher Hardware zusammenarbeiten und schließlich auch noch die parallele Bearbeitung von Prozessen erlauben.

Die Antwort auf diese Herausforderung ist Ada: eine neue Standardsprache, die alle anderen Hochsprachen ablösen soll.

Um das zu erreichen, oder besser gesagt, um zu verhindern, daß Ada auch nur wieder eine Hochsprache von vielen wird, wurden die begabtesten Sprachexperten aus Wissenschaft und Industrie zur Entwicklung herangezogen. Diese Entwickler haben der Sprache Ada viele erprobte und bewährte Elemente anderer Hochsprachen einverleibt, mit dem Ziel, »modern software practices«, das heißt also, moderne Methoden der Software-Entwicklung zu ermöglichen.

Das Ada-Konzept

Die prinzipiellen Eigenschaften von Ada lassen sich in mehreren Gruppen zusammenfassen:

1. Definition von Datentypen, wie:
TAGE_IM_MONAT: INTEGER RANGE 20..31;
erzeugt hier eine Variable mit dem Namen »TAGE_IM_MONAT«, welche nur die ganzzahligen Werte von 20 bis 31 annehmen kann. Damit kann der Programmierer seine eigenen Datentypen erfinden und sie dann zur Benennung von Variablen hernehmen.

Die Datentypen können auch ihre eigenen Werte explizit benennen, wie zum Beispiel:
TYPE MONAT IS (JAN,FEB,MAR, APR);
DER_BESTE_MONAT : MONAT;
Dann kann man schreiben:
IF DER_BESTE_MONAT = APR THEN ....

Diese Eigenschaft von Ada erhöht nich nur die Lesbarkeit, sondern erleichtert es dem Programmierer, die Übersicht zu behalten und erlaubt anderen Leuten, das Programm zu verstehen.

2. Das Konzept der Programmstruktur stellt Befehle wie IF..THEN..ELSE..ENDIF und CASE..WHEN..ENDCASE zur Verfügung, welche zusammen mit der Schleifenbildung jeden logischen Programmfluß ohne GOTO zulassen.

3. Um zu erreichen, daß mehrere Leute Programmteile schreiben können, die später zusammenfügbar sind, hat ein Programm getrennte Merkmale »nach außen« (specification) und »nach innen« (body). Zum Beispiel kann ein Programmierer ein Programm zur Quadratwurzel schreiben. Die »specification« definiert genau, was andere Benutzer wissen müssen (welche Angaben es braucht und welche Angaben es liefert). Im »body« können Methoden und Formeln verwendet werden, die niemanden außer den Autoren selbst interessieren.

4. Ada-Programme bestehen aus »Paketen«, das heißt sie sind blockstrukturiert. In einem Paket sind beliebige Daten und Anweisungen zusammengefaßt, die eine ganz bestimmte Aufgabe erfüllen. Ein Paket kann in verschiedenen Systemen verwendet werden. Es ist letztendlich möglich, Software-Pakete genauso in bestehende Systeme »einzustecken«, wie man das heute mit integrierten Chips macht.

5. Schließlich soll Ada auch Multitasking erlauben. Darunter versteht man die Möglichkeit, mehrere Programmteile gleichzeitig und unabhängig voneinander ablaufen zu lasen und sie nur an ganz bestimmten Stellen zusammenzubinden.

Soviel zur Sprache Ada selbst. Was aber ist der Stand der Entwicklung heute?

Ada wird jetzt gerade eingeführt, aber nicht nur vom Militär und nicht nur in den USA.

Europäische Aktivitäten konzentrieren sich auf Frankreich, Deutschland, Finnland und Großbritannien. Es haben sich schon User-Groups gebildet, Universitäten beschäftigen sich mit Anwendungsstudien und viele Organisatio-

nen bemühen sich um die Entwicklung von Lehr- und Trainingsprogrammen.

Eins haben alle diese Aktivitäten gemeinsam: sie beziehen sich ausschließlich auf große und mittlere Computeranlagen und beschränken auf diese Weise den Zugang zu Ada nur auf den exklusiven Kreis der Profis.

Ada auf dem C 64

In dieser Situation bildet der Ada-Trainingskurs für den C 64 von Data Becker eine hochzulobende Ausnahme, denn er führt erstmalig Ada in die Welt der Home-Computer ein. Der Trainingskurs besteht aus einem Handbuch und einer Programmdiskette, mit deren Hilfe Ada so erklärt wird, daß komplette Programme in dieser Sprache geschrieben werden können.

Die Programmdiskette enthält fünf Programme:
— einen Editor
— einen Syntax-Prüfer
— einen Semantik-Prüfer plus Code-Generator
— einen Assembler (Übersetzer)
— einen Disassembler

Für einen Hobby-Programmierer, der nur Basic kennt, ist das sehr verwirrend. Aber ich bitte zu bedenken, daß auch Basic nicht einfach so abläuft, wenn »RUN« eingetippt wird, sondern daß jede Basic-Zeile vom Computer intern in einen komplizierten Vorgang analysiert werden muß. Dieser Analyse-Vorgang ist in den Commodore-Computern fest eingebaut. Für Ada muß der Trainingskurs einen entsprechenden Übersetzer separat zur Verfügung stellen — eben den oben aufgelisteten Code-Generator und den Assembler.

Außerdem ist jedem Basic-Programmierer geläufig, daß der Rechner merkt, wenn ein Befehl falsch geschrieben worden ist, ein Komma fehlt oder in einer Zeile sonst irgend ein Fehler gemacht worden ist. Die nicht immer verständlichen englischen Fehlermeldungen sind ja oft genug Grund zur Frustration.

Diese Überprüfung besorgt beim Ada-Trainingskurs das Syntax- und das Semantik-Prüfprogramm, welche auf Einhaltung der Rechtschreibung und der Grammatik von Ada achten.

Der Editor steht genau wie bei Basic am Anfang aller Aktivitäten. Bei Basic denken wir nicht lange nach und nehmen es als gegeben hin, daß wie per Tastatur Zeichen, Zahlen und Texte schreiben und eingeben können, daß der Cursor bewegt werden kann und daß uns alle möglichen Steuertasten zur Verfügung stehen. Das eingebaute Betriebssystem stellt uns diese Editor-Funktion zur Verfügung.

Für Ada muß das alles extra gemacht werden und dazu dient das Editor-Programm.

Der Disassembler schließlich ist ein Luxus, der zusätzlich zum Assembler angeboten wird, für ein Ada-Programm aber nicht unbedingt benötigt wird. Soviel sei zur Programmdiskette gesagt.

Meine Meinung über das Handbuch ist zweigeteilt. Einerseits finde ich die Einführung in Ada und die Programmierung sehr gelungen, verständlich und gut lesbar. Die Übungsbeispiele sind klar dokumentiert und erläutert.

Andererseits aber sind die Anweisungen, wie die Übersetzungsprogramme zu bedienen sind, vermischt mit einer sehr detaillierten Beschreibung der Arbeitsweise eben dieser Programme, die nur für Fachleute verständlich ist. Das hat zur Folge, daß der Ada-Anwender, der sich nur für die Sprache selbst interessiert, in Erklärungen ertrinkt, welche für das Verständnis und die Anwendung von Ada allein nicht notwendig sind.

Ich selbst habe mehrere Abende gebraucht, um ein erstes kleines Ada-Programm zum Laufen zu bringen und dafür — ich bin so frei — gebe ich dem Handbuch die Schuld.

Zusätzlich enthalten die Teilprogramme einige kleine Unzulänglichkeiten, auf die ich im folgenden noch eingehe.

Das erste Ada-Programm

Zunächst aber möchte ich beschreiben, wie man zum ersten Ada-Erfolgserlebnis kommt.

Zuerst wird der Editor geladen. Er meldet sich nach kurzer Ladezeit von selbst und muß nicht — wie im Handbuch beschrieben — mit »RUN« gestartet werden.

Der Editor, und noch viel mehr die anderen Programmteile, haben zum Teil ziemlich lange Ladezeiten, die weder im Text noch auf dem Bildschirm angekündigt werden. So ist man häufig im Zweifel, ob man noch warten soll, oder ob das Programm abgestürzt ist, was leider zuweilen auch vorkommt.

Erfreulich, weil leicht verständlich und übersichtlich, sind die verschiedenen Menüs.

Abgesehen von anfänglichen Farbeinstellungen zeigt das Editor-Menü die Befehle für zwei Aktionsgebiete an:
— Schreiben beziehungsweise Ändern des Ada-Programmtextes
— Weiterverarbeiten des Ada-Programms, Ein- und Ausgabe

Beide Befehlssätze bedienen sich der Funktionstasten. Da jede Funktionstaste dadurch zwei Bedeutungen hat, muß der Lernende am Anfang ziemlich oft die Menüs aufrufen, um seinem Gedächtnis auf die Sprünge zu helfen.

Der Editor

Das Schreiben und Ändern verfügt über sehr komfortable Hilfsmittel, wie automatische Zeilennumerierung, Einfügen von Zeilen, vor- und rückwärts.

Ein Programm, auch ein unfertiges, kann mit dem zweiten Befehlssatz ausgedruckt oder auf Diskette abgespeichert werden. Ein abgespeichertes Programm ist von Diskette wieder ladbar. Überhaupt stehen alle gängigen Anweisungen an die Diskettenstation zur Verfügung.

Wenn man schließlich glaubt, daß ein Programm fertig und natürlich fehlerfrei ist, wird es zum Übersetzen geschickt.

Die entsprechende Funktionstaste zaubert neue Anweisungen auf den Bildschirm, welche abfragen, ob man wirklich übersetzen will und wenn ja, ob das Programm zu allererst abgespeichert werden soll. Die erste Frage bietet eine belächelbare, die zweite Frage eine sehr sinnvolle Sicherung gegen Programmverlust.

Die dritte Frage nach einer »Spur« erscheint im Textbuch nicht. Ich habe weder sie noch ihre Auswirkungen irgendwo finden können. Natürlich ist es durch Experimentieren letztlich möglich, den Sinn der »Spur«-Entscheidung herauszutüfteln. Aber hier muß die Frage erlaubt sein, ob das im Sinne eines Lernprogramms ist? Für alle Interessierten: Mit der »Spur«-Option wird der Compiler angewiesen, zusätzliche Textangaben, die Auskunft über die gerade abgearbeiteten Zeilen geben, in den erzeugten Code einzucompilieren. Es handelt sich also um eine Art »Trace«-Funktion für Maschinensprache.

Doch weiter: Immer noch als Teil des Editors erfolgt jetzt eine »lexikalische Analyse«, welche (Zitat) »die einzelnen Worte des Programms erkennen und Worte, die in Ada keinen Sinn ergeben, ausfiltert«.

Bevor mitgeteilt wird, ob diese Analyse erfolgreich war oder Feh- ►


64ER ONLINE

ler aufgespürt wurden, wird erst einmal die Programmdiskette verlangt, von welcher der nächste Programmteil, nämlich der Syntax-Prüfer eingelesen und nach einer längeren angsterfüllten Wartezeit gestartet wird.

Die Syntax-Analyse prüft, ob ein Programm den grammatischen Regeln von Ada entspricht.

Ein »lexikalischer« Fehler (»krocedure« statt »procedure«) führt prompt zu einer Fehlermeldung, gefolgt von der Frage, ob der »Stack« ausgegeben werden soll. Im Handbuch ist zwar, eingebettet in eine ausführliche, an den Fachmann gerichtete Beschreibung der Arbeitsweise des Syntax-Prüfers, diese Ausgabemöglichkeit erwähnt aber nicht, wie sie herbeigeführt wird. Mehrfaches Drücken der RETURN-Taste brachte in der Tat Zahlenreihen auf den Schirm, deren Bedeutung zwar im Detail beschrieben wird, dem Laien aber nicht weiterhilft.

Ausgezeichnet dagegen ist die nachfolgende Fehlerdiskussion, welche die fehlerhafte Zeile, den Fehler und mögliche Abhilfen und Korrekturen aufzeigt.

Die Syntax-Prüfung wird abgebrochen, und es kommt die Aufforderung, wieder die Programmdiskette einzulegen. Es leuchtet natürlich ein, daß von ihr wieder der Editor eingelesen wird, um den Fehler beheben zu können. Was nicht einleuchtet, ist, daß nach dem Laden das Programm hängen bleibt: Absturz!

Bei Entdeckung eines syntaktischen Fehlers (zum Beispiel ein fehlendes Semikolon) läuft dieselbe Prozedur ab, jedoch mit zwei großen Ausnahmen. Zum ersten wird nach der Fehlerdiskussion die Syntax-Prüfung fortgesetzt, um noch das restliche Programm zu analysieren. Zum zweiten stürzt das Programm beim Wiedereinladen des Editors jetzt erfreulicherweise nicht ab. Natürlich kann in beiden Fällen das fehlerhafte Ada-Programm wieder geladen werden. Aber ein Absturz ist halt einfach ärgerlich.

Sind alle lexikalischen und syntaktischen Fehler ausgemerzt, folgt die Aufforderung, von der Programmdiskette den dritten Teil, nämlich den Semantik-Prüfer, einzulesen. Dieser prüft, ob das Programm vom Aufbau her korrekt ist. Zum Beispiel ist eine Anweisung an den Drucker, die das fehlerhafte Wort »Dricker« verwendet, lexikalisch und syntaktisch richtig, wird aber von der semantischen Analyse erwischt.

Trotz eventueller Fehler übersetzt der Code-Generator des Semantik-Prüfers in einem ersten Durchgang das Ada-Programm in ein Assemblerprogramm. Dieses kann entweder mit dem Assembler endgültig in ein Maschinenprogramm übersetzt werden, oder es muß mit dem Editor erst noch korrigiert werden. Für Assemblercode-Spezialisten kann auch das Assemblerprogramm geladen und direkt verbessert werden. Nur eins, die Befolgung der Aufforderung, eine dieser drei Optionen zu benützen, führt unweigerlich zum Absturz. Es hat einige Versuche gekostet, bis als einzige Vorgehensweise feststand, den Computer zuerst durch Ausschalten zurückzusetzen — wahrhaft keine elegante Methode in einem Trainingskurs.

Ist das Ada-Programm endlich fehlerfrei, wird der Assembler geladen. Er fragt nach dem Namen des vorläufigen Assemblerprogramms — und wehe, Sie denken nicht daran, unaufgefordert die Datendiskette einzulegen, auf der das besagte, zu übersetzende Programm gespeichert ist. Diese kleine Unaufmerksamkeit wird wieder mit Absturz des Programms bestraft.

Wenn Sie es aber richtig machen, übersetzt der Assembler in einem zweiten Durchgang das Programm in echten Maschinencode, der mit »RUN« gestartet und sonst wie ein normales Maschinenprogramm behandelt werden kann.

Wie eingangs erwähnt, braucht man den Programmteil Disassembler für Ada nicht, es sei denn, der Kursteilnehmer will, was der Autor empfiehlt, gleich die Gelegenheit nutzen und Maschinensprache lernen. Dann allerdings kommt ihm die ausführliche Beschreibung von Assembler und Disassembler, die unabhängig von Ada auf eigenen Füssen stehen, sehr zugute.

Zusammenfassung

Ich bin begeistert von der Möglichkeit, die Sprache Ada zu lernen, und in ihr auf dem C 64 programmieren zu können.

Die Lehrmethode einer schrittweisen Einführung in die Sprache ist auch für Laien geeignet, die bisher nur in Basic gearbeitet haben.

Die Übersetzungsmethode ist zwangsläufig etwas langwierig, die vielen Diskettenwechsel lassen sich durch den begrenzten Speicher des C 64 nicht vermeiden. Und wenn man einmal den Dreh gefunden hat, ist das nicht mehr störend.

Bis dahin allerdings ist es ein steiniger Weg. Die Programme sind leider nicht laiensicher. Ein Lernprogramm muß einfach dem dümmsten Bedienungsfehler Rechnung tragen und nicht, wie in diesem Trainingskurs, immer wieder zum Absturz führen.

Der begreifliche Stolz des Autors auf die für meine Begriffe fantastische Leistung, die Prüf- und Übersetzungsprogramme geschrieben zu haben, drängt die Beschreibung derselben zu sehr in den Vordergrund. Auch das Argument, daß dadurch wichtige Grundkenntnisse der Assembler- und Maschinensprache erlernbar sind, zieht meines Erachtens nicht. Denn mit dem Trainingskurs — für den stolzen Preis von 198 Mark — will ich Ada lernen und nicht Assemblersprache.

Es wäre Data Becker dringend zu empfehlen, sowohl das Handbuch als auch die Programme zu revidieren und davon eine zweite Version herauszugeben. Die kritisierten Mängel sind alle leicht korrigierbar, wenn man sich die Zeit nimmt und die Bedürfnisse von Laien vor Augen hält.

Dieses ausgezeichnete Unterfangen, die Zukunftssprache Ada den Home-Computern zu eröffnen, darf einfach nicht an ärgerlichen Programmiermängeln und inkonsequenten Erklärungen scheitern.

Ada — für wen?

Der Ada-Trainingskurs kann allen an modernen Programmiersprachen interessierten Anwendern empfohlen werden. Auch wer sich für Compilerbau interessiert, ist mit diesem Kurs gut beraten, da das Handbuch sehr ausführlich auf allgemeine Prinzipien der Compilierung eingeht. Allerdings sollte man die Fähigkeiten dieser Ada-Trainingsversion nicht überschätzen. Der Ada-Trainingskurs eignet sich entschieden besser zum Lernen von Ada als für größere Programmierungsprojekte.

Nicht verschwiegen werden soll auch, daß eine Ada-Version für einen Heimcomputer immer nur die wichtigsten Aspekte der Sprache unterstützen kann. So unterstützt Ada als Trainingskurs beispielsweise kein Multitasking, also keine Parallelverarbeitung bestimmter Programmabschnitte. Doch damit kann man leben. Entscheidend ist die Möglichkeit, eine moderne Sprache wie Ada mit dem C 64 lernen zu können. (Dr. Helmuth Hauck/ev)

Protext - Textprofi mit 80 Zeichen

Einen entscheidenden Schritt in Richtung professionelle Textverarbeitung macht Protext. Wichtigstes Hauptmerkmal: 80 Zeichen ohne Kompromisse.



80 Zeichen am Bildschirm erleichtern die Textverarbeitung

Textverarbeitungsprogramme für den C 64 gibt es inzwischen wie Sand am Meer. Und fast alle haben eines gemeinsam: Will man den Text in seiner endgültigen Form sehen, so muß man ihn ausdrucken. Es gibt bisher keine befriedigende Lösung, mit den beschränkten Fähigkeiten des VIC-Chips eine saubere 80-Zeichen-Darstellung auf dem Bildschirm zu erhalten, um die gesamte Textbreite vor Augen zu haben. Horizontales Scrolling oder softwaremäßig simulierte 80 Zeichen sind nur Kompromisse. Das einzig Wahre ist eine 80-Zeichen-Karte; eine Hardwarelösung also. Doch leider vertragen sich die meisten Textverarbeitungsprogramme nicht mit diesen Karten.

Doch damit ist jetzt Schluß. Zumindest, wenn man es übers Herz bringt, rund 500 Mark auf den Tisch zu blättern und die 80-Zeichen-Karte »80-ZK-plus« sowie das dafür geschriebene Programm »Protext« zu erwerben. Nicht nur die 80 Zeichen sind es, die dem Ganzen das Prädikat »Textprofi« verleihen. Auch die enorme Vielfalt an Funktionen hebt Protext aus der Masse der Textverarbeitungsprogramme heraus.

Befehlsvielfalt

Wer klein anfangen will, der tippt nach dem Laden des Programmes seine Texte so ein, wie er sie später auf dem Papier sehen will und nimmt etwaige Formatierungen und Einrückungen von Hand vor. Dann nur noch zwei Tasten berührt (F1 und A) und schon wird gedruckt. Kurze Zeit später hält man schwarz auf weiß genau das in Händen, was auf dem Monitor stand.

Besonders angenehm bei der Texteingabe ist das Word-Wrapping: Wörter, die nicht mehr in die aktuelle Zeile passen, werden automatisch in die nächste Zeile übernommen. Diese Funktion kann auch abgeschaltet werden. Zur Texteingabe steht ein sehr umfangreicher Befehlssatz zur Verfügung. So lassen sich zum Beispiel einzelne Buchstaben, Wörter, ganze Bereiche oder der gesamte Text löschen. Auch das Gegenteil, das Einfügen von Textstücken ist möglich, die Handhabung aber etwas kompliziert: Hat man den Einfügemodus eingeschaltet, so darf maximal eine Zeile (80 Zeichen) eingefügt werden. Für längere Einfügungen muß der Einfügemodus entsprechend oft ein- und wieder ausgeschaltet werden. In diesem Fall besteht auch die Möglichkeit, einige Leerzeilen zu setzen und dort den Text nachzutragen. Anschließend können alle überflüssigen Leerzeichen aus dem aktuellen Absatz entfernt werden. Das kann über die Optimierungsfunktion automatisch geschehen.

Sehr nützlich sind auch die von Schreibmaschinen bekannten Tabulatoren. Es sind beliebig viele setzbar und es gibt drei verschiedene Typen: normale Text-, Zahlen- und Spaltentabulatoren. Die Zahlentabulatoren helfen, Zahlen untereinander zu setzen, so daß der Dezimalpunkt (oder wahlweise Komma) immer sauber untereinander steht. Mit den Spaltentabulatoren läßt sich der Arbeitsbereich in senkrechte Spalten einteilen, in denen voneinander unabhängig die Texteingabe erfolgt. Da von Spalte zu Spalte gesprungen werden kann, ist ein problemloses Erstellen von Tabellen und Gegenüberstellungen möglich.

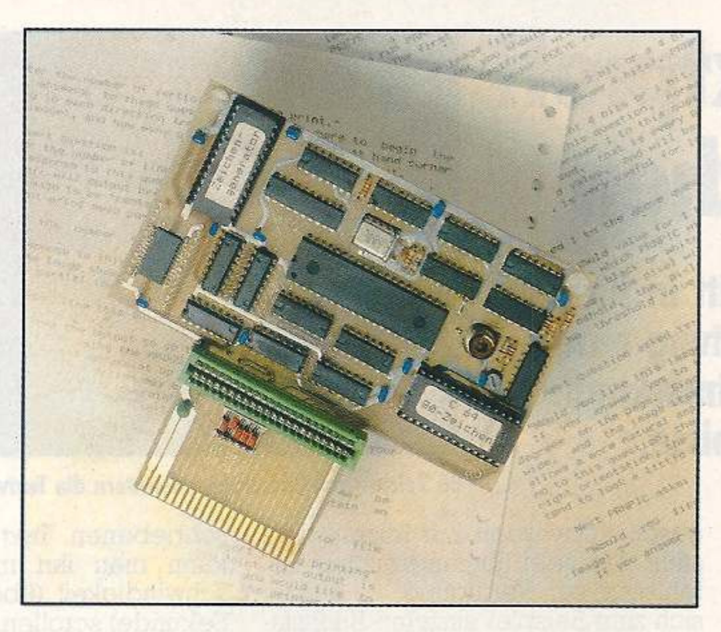
Komplette Textbereiche kann man im Speicher kopieren oder verschieben. Einzelne Buchstabengruppen lassen sich im Text suchen und gegen andere ersetzen. Schließlich besteht die Möglichkeit, mit dem Cursor bestimmte Zeilen direkt anzuspringen.

Sehr praktisch ist die Scroll-Funktion. Möchte man sich einen Überblick über den schon geschriebenen Text verschaffen, so kann man ihn mit enormer Geschwindigkeit (über 20 Zeilen pro Sekunde) scrollen. Die Position des Cursors wird dabei nicht verändert, so daß auch schnelle Richtungswechsel möglich sind. Soweit die Befehle, die ständig benötigt werden. Der Rest des Befehlssatzes dient Spezialanwendungen. Erwähnt werden soll hier nur noch, daß jederzeit beliebige Texte auf Diskette, ohne Textverlust, eingesehen werden können. So auch ein mitgelieferter Hilfstext mit einer Kurzbeschreibung der vorhandenen Befehle.

Exzellente Formatierung

Neben den vielen Befehlen zur Texteingabe gibt es eine Gruppe von Befehlen, die sich erst beim Ausdruck bemerkbar machen: die Anweisungen zur Formatierung des Textes. Neben den fast schon simplen Einstellungen für linken oder rechten Rand kann der Text im Blocksatz, also mit glattem rechten und linken Rand gedruckt werden. Zusätzlich kann Protext bei der Textausgabe lange Wörter trennen. Es werden alle deutschen Trennregeln und die wichtigsten Ausnahmen berücksichtigt. Allerdings können auch unsinnige Trennungen entstehen, die der Benutzer selber abfangen muß, indem er Protext anweist, die entsprechenden Wörter nicht oder anders zu trennen.

Die bedruckbare Länge einer Seite und die Anzahl der Zeilen, die tatsächlich bedruckt werden sollen, können einzeln angegeben werden. Der Text wird dann auf jeder einzelnen Seite automatisch vertikal zentriert. Zusätzlich kann ein Seitenumbruch jederzeit oder bedingt erzwungen werden. Bedingt erzwungen bedeutet, daß ein Seitenumbruch stattfindet, wenn beispielsweise nur die Überschrift eines Absatzes auf die vorherige Seite kom-


Die 80-Zeichenkarte 80-ZK-plus von Decam. Nur mit dieser Erweiterung arbeitet Protext zusammen. Einen ausführlichen Test dazu finden Sie in Ausgabe 4/85.

men würde. Außerdem können Kopf- und Fußzeilen definiert werden, die am Anfang und am Ende jeder Seite gedruckt werden. Eine automatische Seitenzählung ist vorhanden, so daß die Kopf- oder Fußzeile jederzeit die aktuelle Seitenzahl enthalten können. Einspaltige Ein- und Ausrückungen, beispielsweise für Überschriften oder Kapitelnummern, sind ebenso möglich wie die horizontale Zentrierung einzelner Textzeilen.

Die Formatierung kann in einem Text beliebig oft geändert oder zeitweise ganz abgeschaltet werden.

Die Möglichkeiten sind damit noch nicht ganz erschöpft. Diese umfangreiche Übersicht soll aber genügen. Will man die Formatierung überprüfen, kann man den Ausdruck am Bildschirm simulieren. Hier macht sich der Kauf der 80-Zeichen-Karte bezahlt; spart man doch Unmengen von Druckerpapier für Probedrucke.

Jeder Drucker ist geeignet

Bleiben wir beim Thema Drucker, dem wunden Punkt vieler Textverarbeitungsprogramme. Entweder arbeiten sie nur mit wenig Druckern zusammen oder sie nutzen deren Möglichkeiten nicht aus. Protext hat variable Druckertreiber, die sich auf fast jeden erhältlichen Drucker zurechtschneidern lassen. Dadurch kann der Text nicht nur vernünftig ausgedruckt, sondern auch sämtliche Möglichkeiten des Druckers ausgenutzt werden. Das Erstellen des Treibers ist allerdings etwas kompliziert und langwierig, aber nur ein einmaliger Vorgang.

Im Druckertreiber-Menü werden zuerst einmal alle »normalen« Druckerparameter eingestellt. Dazu gehören die Geräte- und Sekundäradresse des Druckers sowie Werte für Zeilenabstand, Papierlänge, Einzelblatteinzug und automatischen Zeilenvorschub.

Als nächstes folgt eine Code-Wandlungs-Tabelle. Jedem der 255 C 64-ASCII-Codes wird ein Druckercode zugeordnet. Diese Tabelle ist äußerst wichtig, da ja der C 64-ASCII-Code, den auch die 80-Zeichen-Karte verwendet, unterschiedlich zum Standard-ASCII-Code der meisten Drucker ist. Es können auch Steuersequenzen angegeben werden, die zu Beginn und am Ende des Textes sowie für jede einzelne Druckzeile gesendet werden.

Weitere Optionen im Treiber-Menü beziehen sich auf die von Protext »serienmäßig« unterstützten Schrifttypen »Unterstreichen« und »Fettdruck«, deren Steuercodes ja auch für jeden Drucker anders sind. Schließlich lassen sich noch Codes für die verschiedensten druckerspezifischen Eigenschaften definieren. So ist beispielsweise das Hoch- und Tiefstellen von Zeichen möglich, indem man entsprechende Steuercode-Sequenzen auf bestimmte Tasten legt (beispielsweise CTRL+S). Auch frei definierbare Zeichen des Druckers können angesprochen werden. Der erstellte Druckertreiber wird auf Diskette gespeichert und kann jederzeit benutzt werden.

Unterstützung von Floppy und Modem

Die Bedienung einer Diskettenstation gestaltet sich ebenso komfortabel wie die des Druckers. Es werden sowohl die 1541 wie auch das 4040-Doppellaufwerk am IEC-Bus unterstützt. Texte können gespeichert und geladen oder in den aktuellen Text eingefügt werden. Zusätzlich ist das Lesen von Daten aus einem sequentiellen File beim Ausdruck von Serienbriefen und das Arbeiten mit Textbausteinen (Modulen) möglich. Längere Texte, die nicht mehr in den Arbeitsspeicher passen, können direkt miteinander verkettet oder in einem Jobfile zusammengefaßt werden. Daß Directories angezeigt und Befehle an die Floppy gesandt werden können, ist fast schon selbstverständlich.

Einen weiteren Pluspunkt handelt sich Protext durch seine Kommunikationsfähigkeiten ein: Terminalbetrieb wird voll unterstützt. Einerseits kann ein Datenverkehr mit Mailboxen aufgenommen werden, indem man Mitteilungen mit Protext erstellt und danach an die Mailbox sendet. Weiterhin können zwei Computer mit Protext über Telefon Texte austauschen. Dazu ist sogar eine automatische Fehlerkorrektur vorhanden.

Nachdem wir die wichtigsten der vorhandenen Funktionen beleuchtet haben, wollen wir Ihnen noch einige Punkte mitteilen, die uns positiv oder negativ aufgefallen sind. Besonders gut gefallen hat uns der mit 40000 Zeichen wirklich großzügig bemessene Arbeitsspeicher. Überzeugen konnte auch die sehr hohe Arbeitsgeschwindigkeit in allen Funktionen. Weniger gefallen haben uns aber die teilweise recht umständliche Bedienung und besonders die exotische Tastaturbelegung. Die Umlaute und das »ß« befinden sich an total unüblichen Stellen, an die man sich nur sehr schwer gewöhnen kann. Dies ist zwar nicht durch das Textprogramm bedingt, sondern durch die 80-Zeichen-Karte, aber trotzdem ein großes Minus. Hier sei nochmals darauf hingewiesen, daß der Betrieb von Protext nur mit der Decam-80-Zeichen-Karte und sonst keiner möglich ist.

Einige Worte noch zum Handbuch: Dies ist ein umfangreiches und hervorragendes Nachschlagewerk. Zum Einarbeiten für einen Textverarbeitungs-Neuling ist es allerdings weniger geeignet, da die Befehle stur in alphabetischer Reihenfolge und nicht gerade didaktisch günstig erklärt werden.

Wer sich nicht scheut, etwas mehr Geld auszugeben und sich die ungewöhnliche Tastaturbelegung gefallen läßt, der wird in Protext einen zuverlässigen und äußerst vielseitigen Partner zur Textverarbeitung finden.

(Boris Schneider/hm)

Info:
Decam, Postfach 1232, 7505 Ettlingen, Tel. 07243/69264
Preis: 198 Mark ohne 80-Zeichen-Karte. 80-ZK-plus: 298 Mark

Logo — die Sprache für Einsteiger

Bekanntgeworden ist Logo in erster Linie als Grafik-Programmiersprache für Anfänger. Spektakuläre Fotos von Vorschulkindern, die in Logo programmieren, gingen bereits durch die Fachpresse. Wir zeigen, was Logo wirklich kann.

Eigentlich sollte man meinen, daß der C 64 auch softwaremäßig endgültig ausgereizt ist. Dennoch gibt es immer wieder Leute, die weder mit dem C 64-Basic, noch mit der massenhaft vorhandenen Software so recht zufrieden sind. Vor allem die von Basic schlecht unterstützte Grafik macht gerade den Einsteigern, die mit Maschinensprache nicht umgehen können, immer wieder Kummer.

Nun bietet Commodore eine Version der Sprache Logo für den C 64 an. Entwickelt vom renommierten Massachusetts Institute of Technology ermöglicht Logo, so das Handbuch, eine sehr einfache und dabei höchst wirkungsvolle Handhabung von Grafik und Listen (Wort- und Zahlenreihen). Grund genug, einmal festzustellen, was Logo wirklich leistet.

Logo wird auf zwei Disketten, einer Programm- und einer Utility-Disk, geliefert. Was nach dem Laden des Programms sofort auffällt, ist die Tatsache, daß Logo im Interpreter-Modus arbeitet, was seine Vor- und Nachteile hat. Einerseits ist es dadurch möglich, im Direktmodus Befehle einzugeben und Fehler sofort festzustellen. So lernt man den richtigen Umgang mit der Sprache und erspart sich das bei Compilersprachen umständliche Verbessern und erneute Compilieren. Andererseits kann ein Interpreter nie die Geschwindigkeit eines compilierten Programmes erreichen.

Logo arbeitet mit der sogenannten Turtle-Grafik, manchem vielleicht bekannt aus der Sendereihe »Mikroelektronik«. Dabei erscheint auf dem Hires-Bildschirm ein Dreieck, eben die Turtle (Schildkröte). Diese kann nun auf dem Bildschirm bewegt werden und hinterläßt dabei eine Spur. Die Befehle FORWARD, BACK, RIGHTTURN, LEFTTURN, denen jeweils eine Zahl folgen muß, lassen die Turtle auf dem Bildschirm herumwandern oder eine Drehung ausführen. Der eingegebene Parameter bestimmt dabei die Anzahl der Schritte. Wer sich schon einmal mit dem Ausrechnen von Koordinaten bei anderen Grafik-Erweiterungen herumgequält hat, wird die Turtle-Methode bestimmt als Erleichterung empfinden.

Die Turtle kann aber auch durch Koordinaten-Eingabe bewegt werden. Mittels der Befehle SETX, SETY und SETXY verändert man den Standpunkt der Turtle, die Befehle XCOR und YCOR liefern die Koordinaten der momentanen Turtle-Position. So bekommt einerseits der Anfänger bereits nach den ersten Tippversuchen sichtbare Ergebnisse (Bild 1), andererseits kann man mit der Turtle bei Anwendung der entsprechenden mathematischen Formeln auch komplizierte Darstellungen realisieren. Die so erstellten Grafiken lassen sich mit SAVEPICT direkt abspeichern, beziehungsweise mit READPICT wieder einlesen.

Das Erstellen von Programmen birgt für den Basic-gewohnten C 64-Benutzer einige angenehme Überraschungen. Um in den Edit-Modus zu gelangen, tippt man TO und einen Programmnamen. Der Name kann frei gewählt werden und dient später zum Aufrufen des Programms. Es empfiehlt sich, Namen zu wählen, die den Zweck des Programms erkennen lassen; durch die besondere Speicherart von Logo hat die Länge des Namens keinen Einfluß auf den Speicherplatzverbrauch. Logo-Programme haben keine Zeilennummern. Sie werden, wenn sie getippt sind, durch das Verlassen des Editors definiert (abgespeichert).

Das grundliegende Konzept von Logo-Programmen ist folgendes: Ein Logo-Programm kann sowohl im Direktmodus, als auch von anderen Programmen aus jederzeit aufgerufen werden. Dadurch ist es möglich, ein Programm in viele Einzelprogramme zu zerlegen und gesondert zu programmieren. Diese Einzelprogramme können unabhängig voneinander abgespeichert werden. Solche »Bausteine« lassen sich auch problemlos in andere Programme einbauen. Ein Logo-Programm besteht also immer aus einer Reihe von Einzelprogrammen.

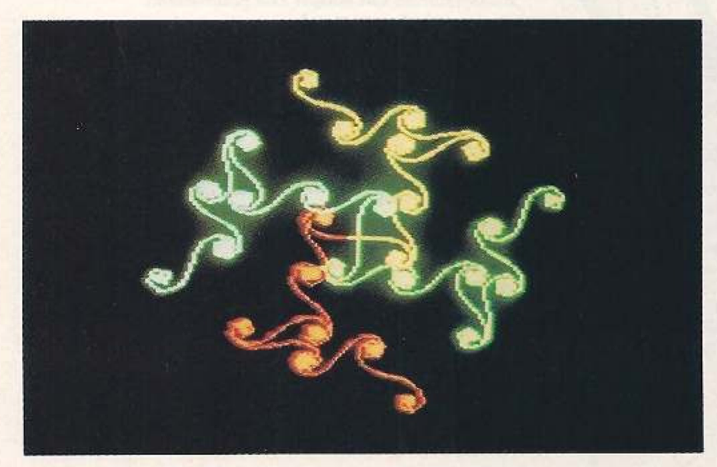


Bild 1. Solche Grafiken bringt man schnell zustande

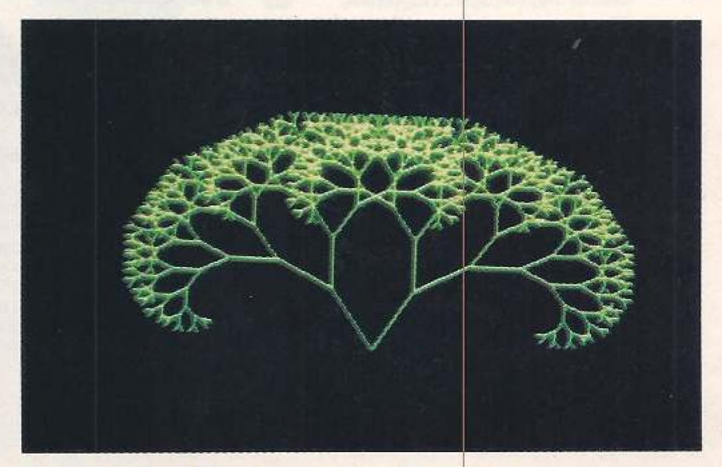


Bild 2. Ein durch Rekursion erzeugter Binärbaum


64ER ONLINE

Die wohl interessanteste Neuerung beim Programmieren ist die Möglichkeit, neue Befehle selbst zu definieren. Nehmen wir an, Sie haben ein Programm zur Errechnung beliebiger Hochzahlen geschrieben, dessen Name »HOCH« ist. Sie haben nun ein anderes Programm, in dem häufig Hochzahlen zu berechnen sind. Es ist möglich, in Ihrem zweiten Programm an den betreffenden Stellen einfach das Programm »HOCH« aufzurufen, das dann die Hochzahl berechnet und an Ihr zweites Programm übergibt. Sie rufen das Hochzahlenprogramm auf, indem Sie »HOCH«, gefolgt von Basis und Exponent als Parameter tippen; der Programmname wird also regelrecht als Befehl benutzt.

Rekursionen

Wer sich im Grafik-Kapitel des Handbuches nach vorne arbeitet, wird bald auf den Begriff »Rekursion« stoßen. Grundlage der Rekursion ist ein Programm, das sich so lange selbst aufruft, bis eine Abbruchbedingung erfüllt ist.

Um nun wieder zum Toplevel, nämlich in den Direktmodus zu gelangen, muß das Programm nach Erfüllung der Abbruchbedingung die Stufen, die es sich hinabgearbeitet hat, wieder hochsteigen. Dabei werden alle Befehle, die nach dem Selbstaufruf stehen, ausgeführt. Sicher ist das zu Anfang kompliziert, aber was sich damit anstellen läßt, zeigt Bild 2. Dabei handelt es sich um einen Binärbaum, vereinfacht gesagt um ein »V«, an dessen Spitze jeweils ein kleineres »V« sitzt etc., bis eine bestimmte Minimallänge erreicht ist. Das Bild wurde mit einem nur wenige Zeilen umfassenden Programm erstellt, das einfach einen Zweig zeichnet und sich jeweils selbst wieder aufruft, um einen weiteren (kleineren) Zweig zu zeichnen.

Ein weiterer Schwerpunkt der Logo-Anwendung liegt in der Listenverwaltung. Listen sind für Logo alle Zahlen, Buchstaben oder Zeichen, die zwischen eckigen Klammern stehen. Logo kann nun mit insgesamt 15 Listenbefehlen einzelne Listen und Wörter miteinander vergleichen, kombinieren und verändern. Wenn man die Befehle miteinander kombiniert, läßt sich schon einiges anstellen: Denkbar sind Grammatik-Übungsprogramme, Adreßdateien mit den unterschiedlichsten Suchkriterien, Text-Adventures oder einfache Dolmetscher-Programme. Die Befehle FIRST, LAST, BUTFIRST, BUTLAST und ITEM ermöglichen beispielsweise ein Herausnehmen einzelner Wörter oder Listenteile aus Listen. Mit den Befehlen SENTENCE und WORD lassen sich Listenteile und Buchstaben zu neuen Wörtern und Sätzen zusammenbauen.

Neben Listenbehandlung und Grafik bietet Logo auch verschiedene Befehle für Musik und Sprites. Der Hersteller macht dabei ausgiebigen Gebrauch von der Möglichkeit, Befehle von Logo aus zu definieren: Für den Umgang mit Sprites sind die Dateien SPRITES und SPRED, für die Musikprogrammierung die Datei MUSIC von der mitgelieferten Utility-Diskette einzulesen. Danach steht dem Benutzer neben den verschiedenen Sprite-Befehlen ein durchschnittlicher Sprite-Editor zur Verfügung. Die entworfenen Kobolde (so werden die Sprites im Logo-Handbuch bezeichnet) können mit SAVESHAPES abgespeichert und mit READSHAPES wieder eingelesen werden. Für Musikprogramme kann man mittels der vordefinierten Befehle Tondauer, Tonhöhe, Attack, Decay, Sustain und Release bestimmen.

Logo — ein vorbildlich dokumentiertes Programm

An dieser Stelle muß ein Wort zur Dokumentation von Logo verloren werden, die wirklich hervorragend ist. Ganz im Gegensatz zur sonst üblichen Commodore-Hauspolitik, nach der der Anwender mit Informationsmengen versorgt wird, die zum Leben zu wenig und zum Sterben zuviel sind, ist diesmal im Lieferumfang ein 351 DIN-A4-Seiten starkes Handbuch enthalten. Es erklärt verständlich und ausführlich die Logo-Befehle; didaktisch gut sind die Programmieraufgaben in jedem Kapitel, die im Anhang gelöst und besprochen werden. Anhand dieses Leitfadens kann man sich ziemlich schnell und gut in Logo einarbeiten. Abgesehen davon, daß sich der Befehl SAVESHAPES nicht in der Datei SPRITES befindet, wie das Handbuch angibt, sondern in der Datei SPRED und die Datei MUSIK sich nur laden läßt, wenn man aus dem »k« ein »c« macht, sind keine Fehler aufgefallen. Ebenfalls nicht selbstverständlich ist die mitgelieferte Utility-Diskette, auf der sich neben zahlreichen Programmierhilfen und Beispielprogrammen, auch ein Drucker- und ein Plotter-Ansteuerungsprogramm und sogar ein Assembler befindet. Da aber der Speicherplatz für Maschinenroutinen mit 447 Byte nicht gerade exzessiv ausgefallen ist und außerdem auch noch von Sprites benutzt wird, so daß man in Sprite-Programme keine Maschinenroutinen einbauen kann, wird der Assembler wohl nicht so häufig benutzt werden.

Neben vielen Vorteilen hat Logo aber auch einige Nachteile. Bedingt durch den Interpreter weist Logo Zeichen- und Ladegeschwindigkeiten auf, die manchmal an das Tempo einer Wanderdüne erinnern. Zum Zeichnen eines Vollkreises (genauer: eines 360-Ecks) brauchte Logo 31 Sekunden; Simons Basic erledigte das in annähernd drei Sekunden. Auch die Ladegeschwindigkeit der Diskette ist mit etwa 2,5 Sekunden pro Block nicht gerade ein neuer Rekord (gerechnet wurde die Zeit vom Ladebefehl bis zur Meldung des Interpreters oder Programmes).

Ein Hindernis für umfangreiche Anwendungen ist wohl der zur Verfügung stehende Speicherplatz. Von ausschlaggebender Bedeutung sind hier die sogenannten »Speicherknoten«. Jeder Gebrauch eines Wortes belegt einen dieser Knoten, und auch ein Programm benötigt eine bestimmte Anzahl davon. In den Arbeitsspeicher kann man 35 Blocks, also etwa 9 KByte laden. Läßt man dann ein Programm laufen, so sind die wenigen übriggebliebenen Knoten durch die Benutzung von Wörtern bald aufgebraucht und Logo meldet »No storage left!« Der vorhandene Speicherplatz mag für Grafikprogramme, selbst für umfangreichere, noch hingehen; ein wirklich brauchbares Dolmetscherprogramm mit ausreichendem Wortschatz beispielsweise, läßt sich so aber kaum erstellen.

Alles in allem ist Logo eine Sprache, die wirklich eine Menge bietet. Turtle-Grafik und Rekursion ermöglichen vielseitige Bildschirmgrafiken, die leicht abzuspeichern sind, und auch die anderen Besonderheiten von Logo sind zum Teil beeindruckend. So kann man die Sprache sicherlich allen empfehlen, die Grafik besser nutzen wollen, sich aber nicht auf Assembler verstehen. Auch die Listenbehandlung in Logo bietet mit etwas Übung interessante Möglichkeiten. Wer dann noch die genannten Einschränkungen in Kauf zu nehmen bereit ist, wird mit Logo bestimmt zufrieden sein.

(Christof Bachmair/ev)

Info: Logo gibt's bei Ihrem Commodore-Händler zum Preis von 159 Mark nur auf Diskette für den Commodore 64.

Assembler ist keine Alchimie — Teil 9

Multiplizieren und Dividieren größerer Zahlen ist weder mit dem Taschenrechner noch in Basic ein Problem. Mit Assembler sieht die Sache anfangs schon nicht mehr so einfach aus. Doch auch diese Hürde wird genommen. Einige Betriebssystemroutinen des C 64 nehmen uns dabei erhebliche Arbeit ab, man muß sie nur kennen.

Einige Versprechen sollen diesmal eingelöst werden: Die restlichen Bit-Verschiebe-Befehle werden Ihnen vorgestellt und auch gleich ein paar Anwendungen wie die 16-Bit-Multiplikation und auch die 16-Bit-Division. Außerdem soll endlich das Programmprojekt weitergeführt werden. Diesmal erzeugen wir einen Hilfsbildschirm und legen ihn abrufbereit unter den oberen ROM-Bereich. Bei der Gelegenheit lernen Sie auch gleich noch ein paar Interpreter-Routinen kennen.

Die restlichen Bit-Verschiebe-Operationen

Da wäre zunächst einmal das Gegenstück zu ASL. Den Befehl haben wir in der letzten Ausgabe kennengelernt. Dort ging es ja um das Nach-Links-Schieben. Jetzt schieben wir nach rechts. LSR heißt der dazu nötige Befehl. Das kommt von »logical shift right« und heißt zu deutsch »logisches Rechtsschieben«. Fragen Sie mich bitte nicht, weshalb »logisches«. Jedenfalls ist LSR ebenso für logische Bitprüfungen geeignet wie ASL.

Mittels LSR wird jedes Bit der adressierten Speicherstelle um einen Platz nach rechts geschoben. An die Stelle des Bit 7 tritt eine Null und Bit 0 wandert in das Carry-Bit (siehe Bild 1).



Bild 1. Wirkung von LSR auf ein Byte

Erinnern Sie sich noch an das dezimale Linksschieben mit ASL aus der letzten Folge? Wir hatten festgestellt, daß jedes Linksschieben einer Dezimalzahl einer Multiplikation mit 10 entspricht. Hier im umgekehrten Fall, also beim Rechtsschieben, muß jedes LSR einer Division durch 10 entsprechen:

25000	wird durch LSR zu	2500
2500	"	250
250	"	25
und so weiter		

Geht man von der Ausgangszahl (25000) aus, dann ergibt sich der erste rechts verschobene Wert durch Division mit

	$10^1 = 10$
der 2. durch	$10^2 = 100$
der 3. durch	$10^3 = 1000$, etc.

Es wird also durch Potenzen der Zahlenbasis 10 geteilt. Haben wir es — wie im Computer — mit Binärzahlen zu tun, deren Basis die 2 ist, dann teilen wir mit jedem LSR durch 2. Je nachdem, wie oft hintereinander das LSR auf eine Zahl ausgeübt wird, teilt man dann durch $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, etc. Das konnte man sich alles ja schon vorstellen, nachdem ASL zur Multiplikation verwendet wurde. Auch hier muß man immer das Carry-Bit abfragen, denn die Division kann ja unter Umständen nicht aufgehen, wie das folgende Beispiel der Division von 3 durch 2 zeigt:

(3) 0000 0011 ergibt durch LSR: 0000 0001 und 1 im Carry-Bit. Das Ergebnis ist schon richtig, nämlich 1. Im Carry steht der Rest dieser Division, die 1. Weil der Rest für manche Berechnungen von Bedeutung ist, muß das Carry-Bit irgendwie erfaßt werden. Wie man das erreicht, lernen wir später noch. Leider ist diese Art der Division mittels LSR nicht so einfach verwendbar wie die Multiplikation mittels ASL. Während man dort durch geschicktes Aufteilen des Faktors ASL auch bei anderen Multiplikatoren als reine Zweierpotenzen anwenden konnte, ist das hier nicht so ohne weiteres möglich. Bei Divisionen geht man deshalb lieber andere Wege. Die zeige ich Ihnen ebenfalls etwas später.

LSR kann auf die gleiche Weise adressiert werden wie ASL:

LSR	auf den Akku bezogen
LSR 6000	absolut
LSR FE	Zeropage-absolut
LSR 6000,X	absolut-X-indiziert
LSR FA,X	Zeropage-absolut-X-indiziert

Im ersten Fall steht das Ergebnis im Akku, in den anderen Fällen in der jeweils adressierten Speicherstelle. Außer der N-Flagge, die in jedem Fall 0 wird, beeinflußt LSR auch die Carry-Flagge und unter Umständen die Z-Flagge. Je nach Adressierungsart liegt LSR als 1-Byte-, 2-Byte- oder 3-Byte-Befehl vor.

Sowohl bei ASL als auch bei LSR hatten wir festgestellt, daß man herausgeschobene Bits, falls sie noch von Bedeutung sind, irgendwie aus dem Carry-Bit (dort sind sie ja gelandet) an einen sinnvollen Ort schaffen muß. Das ist natürlich möglich über eine Befehlskette, in der zunächst das Carry-Bit abgefragt wird:

zum Beispiel:

```
6000    BCC 6007
6002    LDA #01
6004    STA 8000
6007    etc.
```

Wenn das Carry-Bit frei ist, wird alles weitere übersprungen. Wenn da drin etwas aufgetaucht ist, lädt man eine 1 (die ist ja im Carry-Bit) an die benötigte Speicherstelle (hier zum Beispiel 8000). Das kostet aber einige Bytes Speicherplatz und einige Taktzeiten Rechendauer. Außerdem erschwert sich die Programmierung, wenn man eine Zahl öfter verschiebt und dann nach 8000 alle Carry-Inhalte packen will. So kompliziert brauchen wir auch gar nicht zu arbeiten, denn unsere CPU kennt zwei Befehle, die das Bit-Verschieben und das Carry-Verschieben für uns machen. Das sind:

ROL rotate left Linksrotieren
ROR rotate right Rechtsrotieren

Sehen wir uns zunächst mal ROL (Bild 2) an:

Wie bei ASL wandert jedes Bit um eine Position nach links. Das Bit 7 wird dabei in das Carry-Bit verschoben. In Bit 0 gelangt aber hier nicht eine 0 (wie bei ASL), sondern der Inhalt des Carry-Bit (wohlgemerkt der Inhalt, der dort war, bevor dort hinein Bit 7 geschoben wurde). Bevor wir auf den praktischen Nährwert dieses Befehls eingehen, sollen erstmal die Adressierungsmöglichkeiten aufgeführt werden:

ROL		auf den Akku bezogen
ROL	6000	absolut
ROL	FE	Zeropage-absolut
ROL	6000,X	absolut-X-indiziert
ROL	FE,X	Zeropage-absolut-X-indiziert

Je nach Adressierung kann es sich dann wieder um einen 1-Byte- bis 3-Byte-Befehl handeln. Die N-, Z- und natürlich die Carry-Flagge sind beeinflußt und das Ergebnis des Befehls ist im Akku zu finden (erste Adressierungsart) oder in der angesprochenen Speicherstelle.

Wozu das Ganze? Abgesehen von der Möglichkeit, einzelne Bits auf diese Weise ohne Verlust aus einem Byte durch das Carry-Bit herausschieben zu können, um sie Prüfungen zu unterziehen, gibt es noch die Möglichkeit, einen Überlauf bei Rechenoperationen aufzufangen. Erinnern Sie sich an die letzte Folge, wo wir mittels ASL Multiplikationen durchgeführt hatten? Dort kann es unter gewissen Umständen ja leicht geschehen, daß ein Byte für das Ergebnis nicht mehr ausreicht. Wir haben in den Beispielen schon die Überlegung durchgeführt, daß man mittels BCC oder BCS prüfen sollte, ob man eine signifikante Stelle (also eine führende 1) aus dem Byte herausgeschoben hat. Ist das der Fall, dann gibt es zwei Wege:

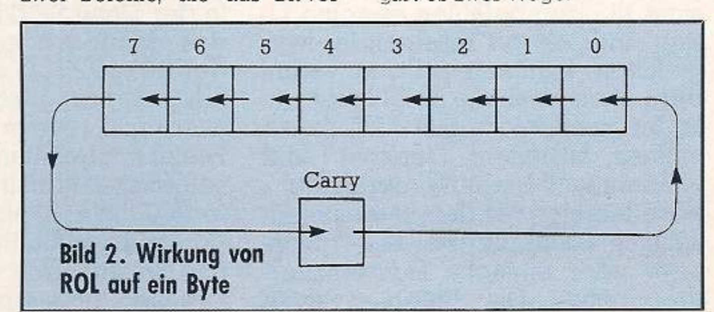


Bild 2. Wirkung von ROL auf ein Byte

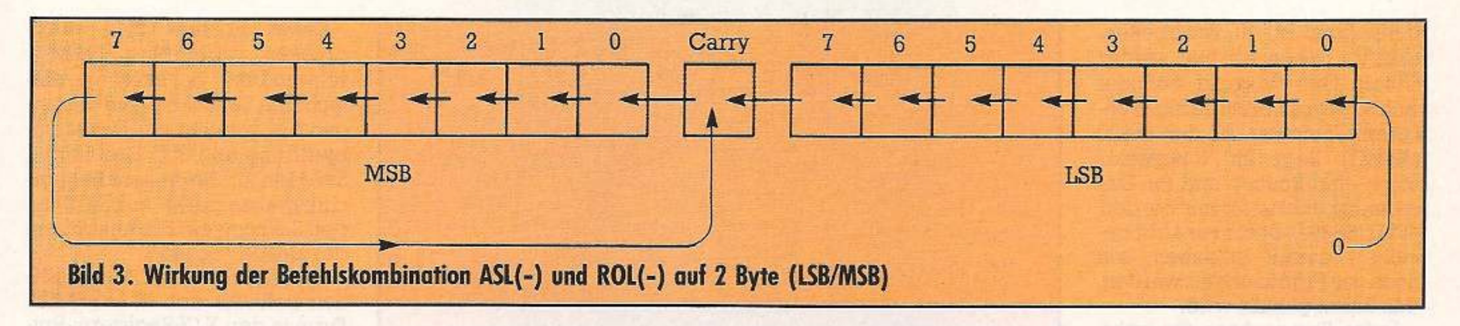


Bild 3. Wirkung der Befehlskombination ASL(-) und ROL(-) auf 2 Byte (LSB/MSB)

1) Man veranlaßt den Ausdruck eines OVERFLOW ERROR, wenn nur 1-Byte-Zahlen zulässig sind, oder
2) man schaltet um auf 2-Byte-Zahlen.

Sehen wir uns das mal an dem Schritt 7 des Beispiels aus der letzten Folge an. Dort hatten wir die Zahl 192 (binär 1100 0000) vorliegen (zum Beispiel in Speicherstelle 7000). Im Computer werden 2-Byte-Integers in der Form LSB/MSB verarbeitet. Wir schaffen also die Speicherstelle für das MSB von 192 in 7001. Jetzt muß dort noch 0 drin stehen. Um bei nochmaliger Multiplikation mit 2 eine 16-Bit-Zahl als Ergebnis zu erhalten, verfährt man wie folgt:

6000	ASL 7000	Damit ist die führende 1 ins Carry-Bit gewandert
6003	BCC 6008	Das setzt man natürlich nur dann ein, wenn man nicht genau weiß, welches Ergebnis zu erwarten ist. Wenn keine 1 ins Carry-Bit gelangte, kann man die nächste Zeile überspringen.
6005	ROL 7001	Damit wurde der Inhalt des Carry-Bit als Bit 0 ins MSB unseres Ergebnisses geschoben.
6008	etc.	

Die Funktion dieser Befehlssequenz können Sie aus Bild 3 entnehmen.

Diesem Befehl werden wir später bei der 16-Bit-Multiplikation und Division noch häufig begegnen.

Sehen wir uns nun noch den letzten der Bit-Verschiebebefehle an: ROR. In Bild 4 ist schematisch gezeigt, wie rotiert wird.

Jedes Bit wandert, wie bei LSR, um eine Stelle nach rechts. Als Bit 7 kommt (im Gegensatz zu LSR) der Inhalt des Carry-Bit herein. Bit 0 wird ins Carry-Bit geschoben. Adressiert werden kann ROR ebenso wie ROL:

ROR	auf den Akku bezogen
ROR 6000	absolut
ROR FE	Zeropage-absolut
ROR 6000,X	absolut-X-indiziert
ROR FE,X	Zeropage-absolut-X-indiziert

Auch für die Byteanzahl, den Ort des Ergebnisses und die Flaggenbeanspruchung gilt dasselbe wie für ROL.

Die Einsatzmöglichkeiten für ROR sind allerings geringer. Bei 16-Bit-Divisionen kann man zwar ROR einsetzen, um einen Unterlauf des MSB ins LSB aufzufangen. Weil man aber meist ohnehin andere Divisionsverfahren verwendet als das oben gezeigte mit LSR, erübrigt sich diese Anwendung in den meisten Fällen. Gut kann man ROR zu Bitprüfungen einsetzen. Das soll im nächsten Abschnitt an einem kleinen Beispiel gezeigt werden.

Zuvor aber noch eine Bemerkung: Wir sind nun durch den Befehlssatz des 6502-Assemblers fast hindurchgedrungen. Es fehlen uns nur noch — wenn ich mich nicht versehen habe — vier Befehle. Die allerdings hängen eng mit dem sogenannten Interrupthandling zusammen, das uns wohl einige Zeit beschäftigen wird.

Schneller Joystick

Vor einiger Zeit (64'er, Ausgabe 2/85) veröffentlichte P. Siepen eine Routine zur Abfrage des Joystickports, die eine interessante Leserbrief-Reaktion hervorrief. M. Hartig sandte nämlich einen Verbesserungsvorschlag, in dem der uns interessierende Befehl ROR die Hauptrolle spielt. Bevor ich die allerdings vorstelle, muß erst noch geklärt werden, was und wie abgefragt wird.

Signale vom Joystick landen in den DATA-Ports A oder B des CIA 1. CIA heißt »Complex Interface Adapter« und ist die Institution unseres Computers, die den Verkehr mit der Außenwelt erlaubt. Wir haben zwei Stück davon (CIA 1 und CIA 2). Je nachdem, in welchen Port der Joystick gesteckt wurde, laufen die Signale in den Registern DC00 oder DC01 (dezimal 56320 oder 56321) ein. Wir nehmen im weiteren mal DC00 an. Die Bits 0 bis 4 beziehen sich auf den Joystick:

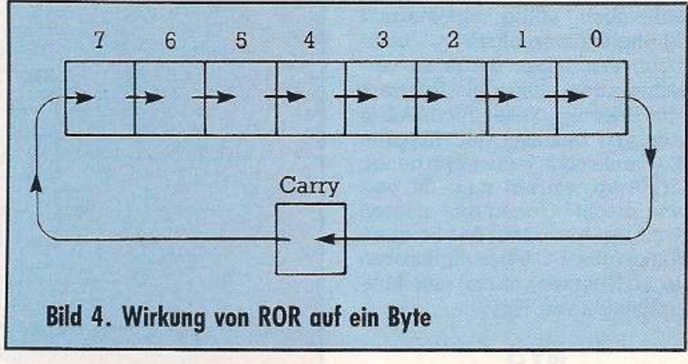


Bild 4. Wirkung von ROR auf ein Byte

Bit 0	oben
Bit 1	unten
Bit 2	links
Bit 3	rechts
Bit 4	Feuerknopf

Wenn keine dieser Möglichkeiten angesprochen ist, enthalten diese Bits den Wert 1. Drückt man beispielsweise den Feuerknopf, dann wechselt der Inhalt von Bit 4 zum Wert 0. Man muß also ständig diese Bits überprüfen und reagieren, sobald eines davon 0 wird. Die Lösung von P. Siepen, diese Abfrage in das Interruptprogramm einzubauen, ist sehr brauchbar. Dadurch hat der Computer die Möglichkeit, trotzdem an anderen Aufgaben weiterzuarbeiten. Wir werden in den nächsten Folgen auf diese Programmiertechnik eingehen. Die Verbesserung von M. Hartig besteht darin, daß er nicht durch CMP-Befehle den Inhalt von DC00 prüft (was Zeit und auch Speicherplatz kostet), sondern mittels ROR Bit für Bit nach rechts in das Carry-Bit schiebt und dieses dann mit BCC abfragt. Sobald die Carry-Flagge nämlich frei ist, ist die zu dem Bit gehörige Joystickfunktion gefragt.

Nun die Abfrageroutine:

LDA DC00	Inhalt des DATA-Port A in den Akku
ROR	Durch Rechtsrotieren wird Bit 0 in die Carry-Flagge geschoben.
BCC Oben	Wenn die Carry-Flagge nicht gesetzt ist, war Bit 0 eine Null, also die Joystickfunktion »Oben« gefordert, zu deren Bearbeitung hier verzweigt werden kann.
ROR	Das nächste Rechtsrotieren schiebt Bit 1 in die Carry-Flagge.
BCC Unten	Auch hier wieder Abzweigen zur Bearbeitung von »Unten«, wenn Bit 1 nicht gesetzt war.
ROR	Bit 2 ins Carry-Bit
BCC Links	und bearbeiten, wenn nicht gesetzt
ROR	Bit 3 in Carry-Flagge
BCC Rechts	und verzweigen wenn Bit 3 Null war
ROR	zu guter Letzt kommt noch Bit 4 ins Carry-Bit
BCC Fire	und kann bearbeitet werden, wenn es Null war.
..... weitere Bearbeitung, wenn keine Joystickfunktion	

Der Vorteil dieser nur 18 Byte langen Unterroutine liegt in ihrer Schnelligkeit: Sie braucht nur 24 Taktzyklen, wenn nicht verzweigt wird, beziehungsweise 25, wenn verzweigt wird. Natürlich wäre anstelle von ROR auch die Verwendung von LSR möglich gewesen, denn die herausgeschobenen Bits werden nicht mehr benötigt. Im Falle, daß man nach einer solchen Abfrage wieder den Ausgangszustand des Akku oder der Speicherstelle herstellen will, muß man eine entsprechende Anzahl ROR-Anweisungen anschließen, bis Bit 0 wieder in seine Ausgangslage rotiert ist.

Die 16-Bit-Multiplikation

Wir haben in der letzten und in dieser Folge gelernt, wie man 8-Bit-Zahlen miteinander malnehmen kann um 8- oder 16-Bit-Zahlen zu erhalten. Dabei ist unbefriedigend, daß man sich über jede Zahl Gedanken machen muß, wie man sie am besten multipliziert. Was fehlt, ist ein allgemein gültiges Programm, das in der Lage ist, jede Zahlenkombination (solange es sich um 2-Byte-Integers handelt und das Ergebnis als 16-Bit-Zahl

darstellbar ist) zu verarbeiten. Und da haben wir mal wieder Glück: Gut versteckt befindet sich so etwas bereits fertig in unserem Computer. Ab dez. 45900 ($B34C) liegt im Interpreter solch eine Routine und ihr Einsprungspunkt ist für uns bei dez. 45911 ($B357). Bevor wir aber detailliert darauf eingehen, soll noch das Prinzip erklärt werden, das dabei genutzt wird.

Jeden Tag rechnen Sie wahrscheinlich völlig automatisch Multiplikationsaufgaben, ohne noch Gedanken daran zu verschwenden, wieviel Schweiß das Erlernen dieser Technik früher mal gekostet hat. Könnten Sie heute noch jemandem genau erklären, warum man da was wie macht? Genau das müssen wir aber tun, damit der Binärautomat (unser C 64) multiplizieren lernt. Nehmen wir mal eine Multiplikation von 16x15:

```
16 x 15
 16
  80
 240
```

Daß wir nicht so genau wissen, was wir da tun, liegt am ziemlich komplizierten Zehnersystem. Damit das alles einfacher und überschaubarer wird, wechseln wir mal ins Binärsystem: 16 = 10000, 15 = 1111. Die Aufgabe sieht dann so aus:

```
10000 * 1111
10000
 10000
  10000
   10000
11110000
```

Jetzt wird schon deutlicher, was wir getan haben. Der Faktor auf der rechten Seite wurde vom MSB an Bit für Bit durchgesehen. Jedesmal, wenn wir auf eine 1 gestoßen sind (hier waren nur Einsen), haben wir den links stehenden Faktor notiert. Dabei sind wir von mal zu mal um eine Stelle nach rechts gerückt, was zu tun hat mit dem Stellenwert des im rechten Faktor gerade betrachteten Bits. Das geschah so lange, bis alle Bits des rechten Faktors durchgearbeitet waren. Die sich auf diese Weise ergebene Kolonne wird dann addiert und führt zum Ergebnis. Vergleichen Sie, 240 ist wirklich binär 1111 0000.

Genauso wie hier beschrieben, arbeitet das Multiplikationsprogramm. Ein Unterschied tritt auf, nämlich daß nicht bis zum Schluß mit der Addition gewartet, sondern jede neue Zwischenzahl sofort addiert wird. Bild 5a zeigt die Beschreibung der Interpreterroutine:

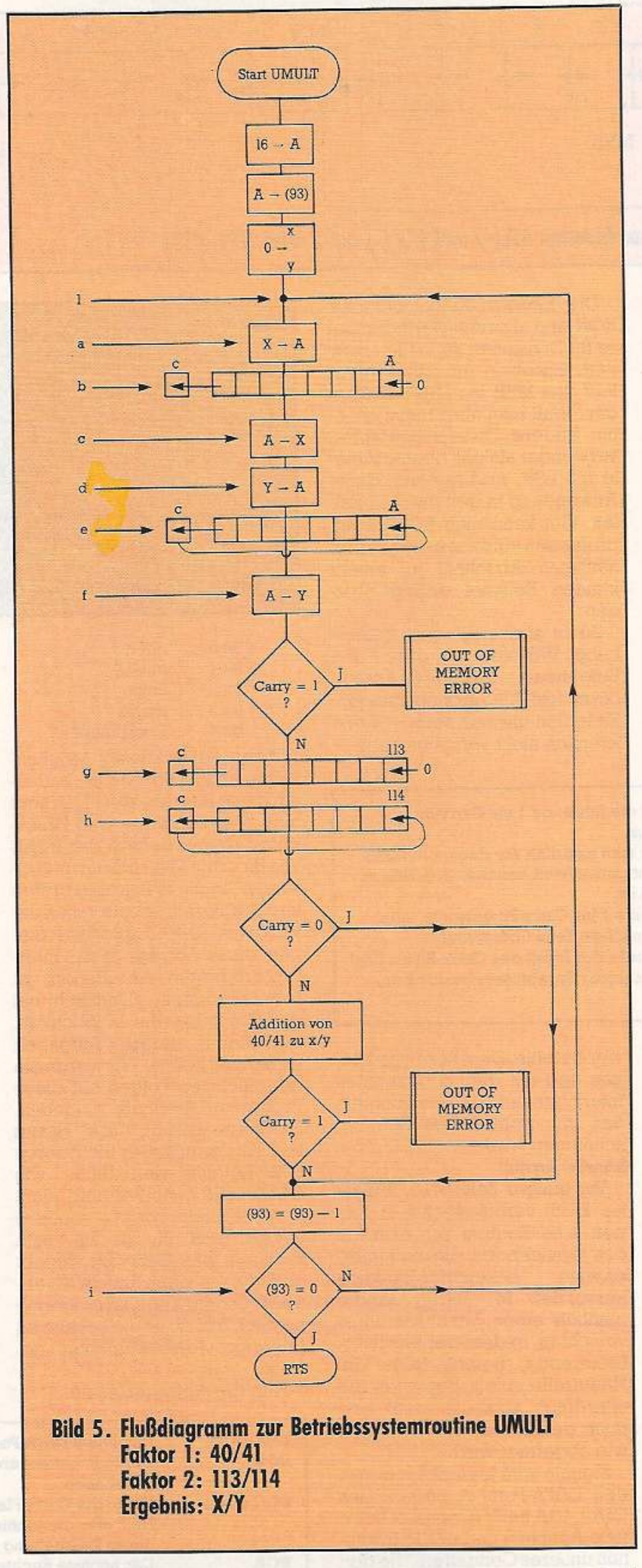


Bild 5. Flußdiagramm zur Betriebssystemroutine UMULT
Faktor 1: 40/41
Faktor 2: 113/114
Ergebnis: X/Y

Name	**UMULT**
Zweck	Multiplikation zweier 16-Bit-Zahlen
Adresse	$B357 dez. 45911
Vorbereitungen	Faktor 1 in $28/29
	Faktor 2 in $71/72
Speicherstellen	$28/29,$71/72, $5D
Register	Akku, X- und Y-Register
Stapelbedarf	keiner

Bild 5a. Die Interpreterroutine UMULT

Diese Routine hier abzudrucken, wäre reine Platzverschwendung. Schalten Sie einfach den SMON ein und verlangen Sie von ihm ein Disassemblerlisting ab B357. Dort haben Sie dann für die weitere Besprechung alles parat. In Bild 5 finden Sie noch ein Flußdiagramm der UMULT-Routine.

Das Ergebnis der Multiplikation befindet sich in LSB/MSB-Form in den X/Y-Registern. Programm und Flußdiagramm wollen wir an einem Beispiel nachspielen. Dazu sollen die beiden Zahlen 321 und 65 (binär 0000 0001 0100 0001 und 0100 0001) miteinander multipliziert werden, was bekanntlich 20865 (binär 0101 0001 1000 0001) ergibt. Was Ihnen im Bild 6 als undurchdringlicher Bit-Dschungel entgegenstrahlt, ist das schrittweise Verfolgen des Programms in Computerformat, also binär.

In Bild 6 sind die Speicheradressen alle dezimal angegeben. Dort finden Sie zunächst die Ausgangslage. In Speicherstelle 40/41 steht die ganze Operation über unverändert die Zahl 321. In 113/114 finden Sie (wegen des LSB/MSB-Formates umgedreht als 114/113) unseren Faktor 65. Akku und Speicherstelle 93 stehen auf 16, dem Bitzähler. In das X- und Y-Register wurde eine Null eingelesen. Im Flußdiagramm ist diese Situation mit einer 1 gekennzeichnet. Ganz unten im Diagramm sehen Sie, daß der Bitzähler 93 erniedrigt und danach geprüft wird, ob er schon gleich Null sei. Daraus folgt, daß die große Schleife 16mal durchlaufen wird. Den ersten Durchlauf (gekennzeichnet durch kleine Buchstaben) verfolgen wir im einzelnen.

a) X-Register wird zur Bearbeitung in den Akku geschoben.

b) Mittels ASL wird das Bit 7 in die Carry-Flagge geschoben, was einen Carry-Inhalt von 0 bewirkt.

c) Der solchermaßen bearbeitete Akku-Inhalt (der sich hier nicht weiter verändert hat) geht wieder zurück ins X-Register.

d) Nun ist das Y-Register zur Bearbeitung dran. Es gelangt in den Akku.

e) Mittels ROL wandert nun das MSB des X-Registers aus dem Carry-Bit in die 0-Bit-Position des Akku

f) und alles zusammen wieder ins Y-Register. Insgesamt wird dadurch die 16-Bit-Zahl im X/Y-Register um eine Stellenzahl erhöht, was der Vorbereitung zur Addition dient. (Erinnern Sie sich bitte: Die Kolonne der Einzelergebnisse wird ja addiert). Im Diagramm (ohne Buchstabenkennzeichnung) schließt sich hier noch eine Prüfung auf einen eventuellen Überlauf an, der dann mit einer Fehlermeldung beantwortet wird.

	114	113	Y	X	A	93	C	
I	00000000	01000001	00000000	00000000	00010000	00010000	—	Ausgangslage
a				→	00000000		—	
b					00000000		0	
c				00000000	—		0	
d				⟶	00000000		0	
e					00000000		0	
f			00000000	⟵			0	
g		10000010					0	
h	00000000						0	Ende 1.
i	00000000	10000010	00000000	00000000	00000000	00001111	0	Schleife
II	00000001	00000100	00000000	00000000	00000000	00001110	1, 0	Ende 2. Schleife
III	00000010	00001000	00000000	00000000	00000000	00001101	0	Ende 3. Schleife
IV	00000100	00010000	00000000	00000000	00000000	00001100	0	Ende 4. Schleife
V	00001000	00100000	00000000	00000000	00000000	00001011	0	Ende 5. Schleife
VI	00010000	01000000	00000000	00000000	00000000	00001010	0	Ende 6. Schleife
VII	00100000	10000000	00000000	00000000	00000000	00001001	0	Ende 7. Schleife
VIII	01000001	00000000	00000000	00000000	00000000	00001000	1, 0	Ende 8. Schleife
IX	10000010	00000000	00000000	00000000	00000000	00000111	0	Ende 9. Schleife
X	00000100	00000000	00000001	01000001	00000001	00000110	1, 0	Ende 10. Schleife
XI	00001000	00000000	00000010	10000010	00000010	00000101	0	Ende 11. Schleife
XII	00010000	00000000	00000101	00000100	00000101	00000100	1, 0	Ende 12. Schleife
XIII	00100000	00000000	00001010	00001000	00001010	00000011	0	Ende 13. Schleife
XIV	01000000	00000000	00010100	00010000	00010100	00000010	0	Ende 14. Schleife
XV	10000000	00000000	00101000	00100000	00101000	00000001	0	Ende 15. Schleife
XVIa,		00000000	01010000	01000000	01010000		0	
b,	00000000	00000000	01010001	10000001	01010001	00000000	1, 0	Ende 16. Schleife

Bild 6. UMULT am Beispiel der Multiplikation 321 x 65 = 20865

g) Nun wird das MSB der Speicherstelle 113 nach links ins Carry geschoben. Das ist auch hier noch eine Null.

h) Anschließend wandert dieser Carry-Inhalt als Bit 0 in Speicherstelle 114. Bit 7 von 114 landet dafür im Carry. Auch hier wird auf diese Weise die ganze 16-Bit-Zahl 113/114 um ein Bit nach links geschoben und im nächsten Schritt — im Flußdiagramm wieder ohne Buchstabe — geprüft, ob da eine 1 oder eine 0 ins Carry-Bit geshiftet wurde. Wenn lediglich eine Null auftrat — wie hier —, dann springt das Programm sofort zum Herabzählen des Bitzählers 93. Tritt aber eine 1 auf, dann addiert sich der Inhalt von 40/41 zu X/Y.

i) Hier wird der Zustand der betroffenen Speicherstellen und Register nach dem ersten Schleifendurchlauf gezeigt.

Römisch II bis XVI zeigen nun jeweils den Zustand nach dem 2. bis 16. Durcharbeiten der großen Schleife. Wenn Sie verstehen möchten, was da passiert, sollten Sie versuchen, Bild 6 nur als Kontrolle zu verwenden und ansonsten mal selbst alle Schritte nachzuvollziehen.

16-Bit-Division

Beim umgekehrten Weg, nämlich der Teilung von zwei 16-Bit-Zahlen, haben wir nicht so viel Glück: Ich konnte keine derartige Routine im Interpreter entdecken. Nun gibt es aber fast in jedem Lehrbuch der Maschinensprache die Vorstellung eines solchen Programms, so daß man sich das schönste aussuchen kann. Das Prinzip ist auch da dasselbe, wie wir es von der normalen Division gewohnt sind: Der Divisor wird Schritt für Schritt vom Dividenden abgezogen. In der Literatur [1] fand ich eine sehr kurze Routine, die ich Ihnen leicht modifiziert als Programm 1 vorstellen will.

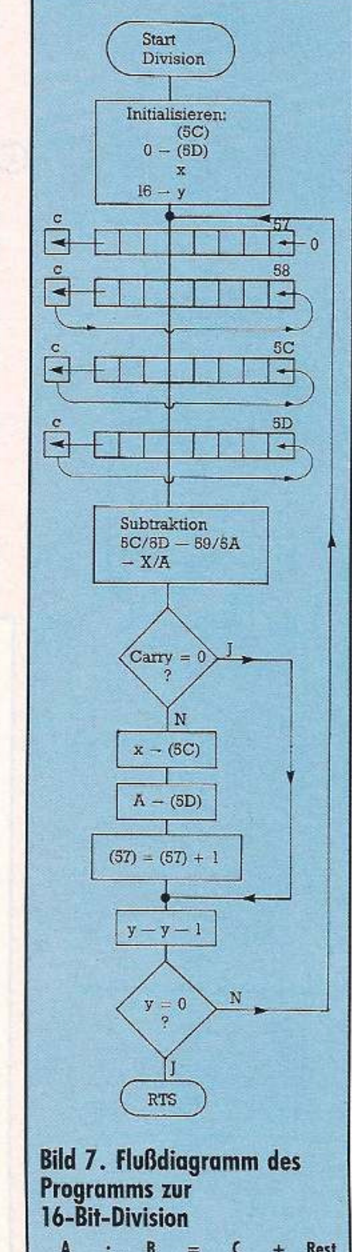


Bild 7. Flußdiagramm des Programms zur 16-Bit-Division

A : B = C + Rest
57/58 59/5a 57/58 5c/5d

In Bild 7 ist ein Flußdiagramm dieser Routine gezeigt und in Bild 8 lacht Ihnen wieder das Bit-Gewirr entgegen, das Sie schon von der Multiplikation her kennen, hier aber für die Division.

Damit Sie wissen, wo was hinein- oder herauskommt:

A	: B	= C	+ Rest
↑	↑	↑	↑
$57/58	$59/5A	$57/58	$5C/5D

An dem folgenden Beispiel soll der Programmverlauf getestet werden: Wir teilen 20867 durch 321. Dabei kommt nach Adam Riese heraus: 65, Rest 2.

In folgender Weise wird in die Speicherzellen die Aufgabe eingespeist:

20867	$57	1000	0011	LSB
	$58	0101	0001	MSB
321	$59	0100	0001	LSB
	$5A	0000	0001	MSB

Als Ergebnis findet man dann:

65	$57	0100	0001	LSB
	$58	0000	0000	MSB
Rest 2	$5C	0000	0010	LSB
	$5D	0000	0000	MSB

Als Bit-Zähler dient hier das Y-Register.

b) Erstes Linksschieben des LSB mittels ASL. Dabei gelangt die 1 in das Carry-Bit.

c) Hineinrotieren der 1 aus dem Carry in das MSB mittels ROL.

d), e) Linksrotieren der 16-Bit-Zahl in $5C/5D, die jetzt noch 0 ist.

f) Situation am Ende der ersten Schleife. Der Bitzähler ist um 1 reduziert.

Im folgenden wird dann jeweils die Situation am Ende der Schleife gezeigt. Beim Berechnen der Differenz muß jeweils darauf geachtet werden, daß die Subtraktion einer Zahl als Addition des Zweierkomplements ausgeführt wird. Das haben wir in den Folgen 3 und 4 der Serie kennengelernt. Allerdings muß an dieser Stelle nochmal gesagt werden, daß die 1, die zum Einerkomplement hinzuaddiert wird, um das Zweierkomplement zu erhalten, das gesetzte Carry-Bit ist. Nun dürfte es für Sie eigentlich keine Probleme mehr geben, was das Nachvollziehen der Divisionsroutine betrifft.

Damit dürfen wir getrost die 16-Bit-Arithmetik abschließen. Alle vier Grundrechnungsarten können Sie jetzt programmieren. Weitere Rechenarten, wie Potenzieren, das Ziehen von Wurzeln, Logarithmen etc. bedingen ohnehin, daß die Argumente oder Ergebnisse keine Integerzahlen sind. Hier werden wir dann mit Fließkommaarithmetik arbeiten und den dazu vorgesehenen Interpreterroutinen.

Das Programmprojekt wird fortgeführt

Im 6. Teil dieser Serie haben wir ein Projekt gestartet, das dort eine Kopfzeile rückholbar unter den oberen ROM-Bereich verschob. Unser Wissen ist seither gestiegen und damit auch unsere Ansprüche. Eine Kopfzeile reicht nicht mehr, jetzt soll es ein ganzer Hilfsbildschirm sein, den wir erst in aller Ruhe erstellen wollen, um ihn dann jederzeit abrufbar unter das Betriebssystem zu packen. Den Aufruf wollen wir wieder mit der USR-Funktion steuern. Diesmal soll aber so programmiert werden, daß der Hilfsbildschirm erhalten bleibt, man ihn also mehrfach einblenden kann. Über die Nützlichkeit einer solchen Routine braucht man sicherlich nicht viele Worte zu verlieren: Denken Sie da nur mal an Programme, die irgendwelche Tasten mit besonderen Funktionen belegen, für die Sie eine Gedächtnisstütze brauchen, oder ...

Als Programm 2 ist ein kleines Demo-Programm abgedruckt, welches zuerst einen Bildschirm erstellt, dann die Routine »Ver-

	58	57	5A	59	5D	5C	A	X	Y	C	
I a	0 1 0 1 0 0 0 1	1 0 0 0 0 0 1 1	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	—	0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 0	—	Ausgangslage n. Init.
b		0 0 0 0 0 1 1 0								1	1. Linksschieben
c	1 0 1 0 0 0 1 1									0	2. Linksschieben
d, e					0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0				0	3./4. Linksschieben
f	1 0 1 0 0 0 1 1	0 0 0 0 0 1 1 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 0	1 0 1 1 1 1 1 1	0 0 0 0 1 1 1 1	1, 0	Ende der 1. Schleife
II	0 1 0 0 0 1 1 0	0 0 0 0 1 1 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	1 1 1 1 1 1 1 0	1 1 0 0 0 0 0 0	0 0 0 0 1 1 1 0	1, 0, 1, 0	Ende der 2. Schleife
III	1 0 0 0 1 1 0 0	0 0 0 1 1 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	1 1 1 1 1 1 1 0	1 1 0 0 0 0 0 1	0 0 0 0 1 1 0 1	1, 0	Ende der 3. Schleife
IV	0 0 0 1 1 0 0 0	0 0 1 1 0 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 1	1 1 1 1 1 1 1 0	1 1 0 0 0 1 0 0	0 0 0 0 1 1 0 0	1, 0, 1, 0	Ende der 4. Schleife
V	0 0 1 1 0 0 0 0	0 1 1 0 0 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 1 0 1 0	1 1 1 1 1 1 1 0	1 1 0 0 1 0 0 1	0 0 0 0 1 0 1 1	1, 0	Ende der 5. Schleife
VI	0 1 1 0 0 0 0 0	1 1 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 1 0 1 0 0	1 1 1 1 1 1 1 0	1 1 0 1 0 0 1 1	0 0 0 0 1 0 1 0	1, 0	Ende der 6. Schleife
VII	1 1 0 0 0 0 0 1	1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 1 0 1 0 0 0	1 1 1 1 1 1 1 0	1 1 1 0 0 1 1 1	0 0 0 0 1 0 0 1	1, 0, 1, 0	Ende der 7. Schleife
VIII	1 0 0 0 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 1 0 1 0 0 0 1	1 1 1 1 1 1 1 1	0 0 0 1 0 0 0 0	0 0 0 0 1 0 0 0	1,1,0,1,0	Ende der 8. Schleife
IX	0 0 0 0 0 1 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	1 0 1 0 0 0 1 1	1 1 1 1 1 1 1 1	0 1 1 0 0 0 1 0	0 0 0 0 0 1 1 1	1, 0, 1, 0	Ende der 9. Schleife
X a	0 0 0 0 1 1 0 0	0 0 0 0 0 0 0 0			0 0 0 0 0 0 0 1	0 1 0 0 0 1 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 1		1, 0, 1, 1	
b	0 0 0 0 1 1 0 0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 1	0 0 0 0 0 1 1 0		Ende der 10. Schleife
XI	0 0 0 1 1 0 0 0	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 1 0 1 0	1 1 1 1 1 1 1 0	1 1 0 0 1 0 0 1	0 0 0 0 0 1 0 1	1, 0	Ende der 11. Schleife
XII	0 0 1 1 0 0 0 0	0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 1 0 1 0 0	1 1 1 1 1 1 1 0	1 1 0 1 0 0 1 1	0 0 0 0 0 1 0 0	1, 0	Ende der 12. Schleife
XIII	0 1 1 0 0 0 0 0	0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 1 0 1 0 0 0	1 1 1 1 1 1 1 0	1 1 1 0 0 1 1 1	0 0 0 0 0 0 1 1	1, 0	Ende der 13. Schleife
XIV	1 1 0 0 0 0 0 0	0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 1 0 1 0 0 0 0	1 1 1 1 1 1 1 1	0 0 0 0 1 1 1 1	0 0 0 0 0 0 1 0	1, 0	Ende der 14. Schleife
XV	1 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	1 0 1 0 0 0 0 1	1 1 1 1 1 1 1 1	0 1 1 0 0 0 0 0	0 0 0 0 0 0 0 1	1, 0, 1, 0	Ende der 15. Schleife
XVIa,	0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0			0 0 0 0 0 0 0 1	0 0 0 0 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0		1,1,0,1,1	
b,	0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 1	0 1 0 0 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0		Ende der 16. Schleife = Endlage

Bild 8. 16-Bit-Division Schritt für Schritt am Beispiel 20867:321=65 Rest 2

schieben« aufruft, den Bildschirm löscht und neu beschreibt und schließlich mit einem weiteren USR den alten Bildschirm einblendet (vorher Programm 3 und 4 laden).

Von nun an können Sie immer — auch im Direktmodus — durch ein USR-Kommando diesen Bildschirm abbilden. Zum Programm in der Folge 6 sind noch zwei Dinge zu bemerken, die hier geändert werden sollen. Erstens eine Frage: Ist Ihnen der Computer mal abgestürzt beim Aufruf des Programms? Die Wahrscheinlichkeit dafür ist ungefähr 1 : 60, wenn nämlich ein Interrupt stattfindet, während die Speicherstelle 1 geändert wird. Obwohl wir erst in den nächsten Folgen auf Interrupts eingehen werden, wollen wir die Wahrscheinlichkeit für so einen Absturz auf Null reduzieren. Eine andere Sache ist der Ort, an dem sich das Programm befand. Es hat sich nämlich herausgestellt, daß anscheinend die Nutzung dieses dort gewählten Speicherbereichs nicht ganz so problemlos ist. Bei einigen Anrufen wurde mir erzählt, daß zumindest der Anfang ab $02A7 bei bestimmten Konstellationen überschrieben wird. Deswegen packen wir unser Programm ganz unkonventionell nach $6000, von wo Sie es — das beherrschen Sie ja mit dem SMON inzwischen sicher — dorthin schieben können, wo es Ihnen gefällt. Allerdings müssen dann auch die USR-Adressen geändert werden. Aber auch das dürfte für Sie inzwischen kein Problem mehr sein.

Um diese immerhin schon 2000 Byte (1000 für den Bildschirm und nochmal 1000 für das Farb-RAM) zu verschieben, bedienen wir uns einer Interpreter-Routine, die seit Ausgabe 3/85 des 64'er auch beim Checksummer verwendet wird — der Blockverschiebe-Routine (Bild 9a).

Wieder besteht unser Programm aus zwei Teilen. Im ersten wird der aktuelle Bildschirm nach oben geschoben. Dieser Teil speist lediglich zuerst die Adressen des Bildschirms und des Betriebssystem-ROM in die Abholspeicherstellen der danach aufgerufenen Routine BLTUC und wiederholt diesen Vorgang für die Bildschirmfarbspeicheradressen. Danach verstellen wir noch den USR-Vektor und kehren mit TRS ins Basic-Programm zurück (siehe Programm 3).

Komplexer ist der zweite Teil. Um nämlich die Informationen unter dem ROM lesen zu können, muß dieses ausgeschaltet werden. Leider läßt sich das Betriebssystem-ROM nur zusammen mit dem Basic-Interpreter ausschalten. $A3BF ist aber eine Interpreter-Routine! Da bleibt uns nichts anderes übrig, als diese Routine in unser Programm einzubauen, was uns die Gelegenheit gibt, sie uns mal etwas anzusehen. Als Bild 9 ist sie im Flußdiagramm abgebildet.

Programm 4 zeigt den zweiten Teil unseres Hilfsbildschirm-Programms.

Von $6040 an, wohin wir am Ende des ersten Teils den USR-Vektor gerichtet haben, wird zunächst wieder Quell- und Zielbereich in den Abholspeicherstellen spezifiziert und jeweils danach zuerst für den Bildschirm, dann für das Farb-RAM, das übernommene Unterprogramm angesprungen. Ab $6077 liegt dann das modifizierte

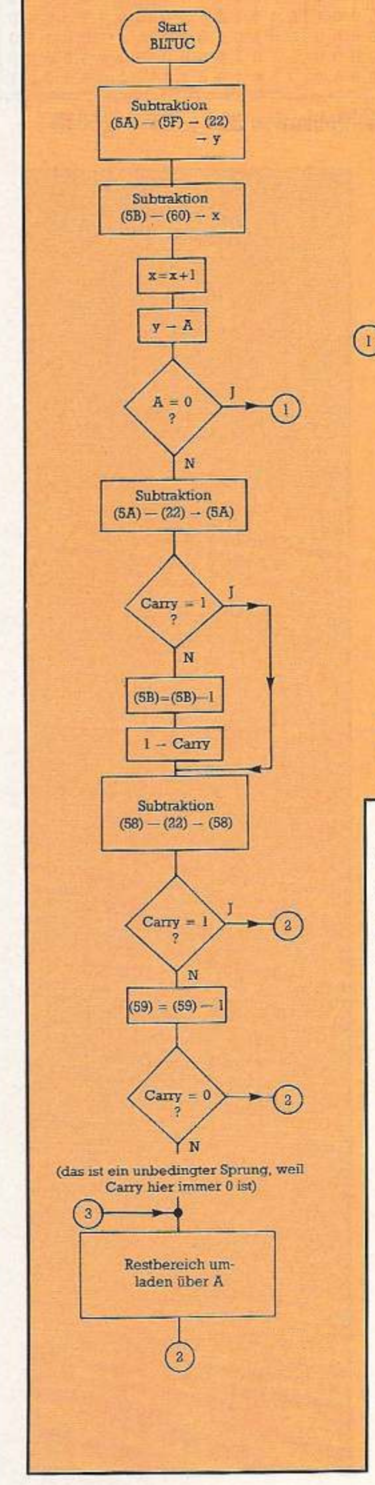


Bild 9. Flußdiagramm zur Betriebsroutine BLTUC

Unterprogramm. Die Befehle SEI und CLI gehören zu den wenigen, die Sie erst noch kennenlernen. Sie sind es, die die Absturzwahrscheinlichkeit auf Null bringen. Jedenfalls wird zuerst das ROM aus und dafür RAM eingeschaltet. Ab $607F bis $60B9 befindet sich die Interpreter-Routine **BLTUC**. Darin wird zunächst die Länge des zu verschiebenden Bereichs berechnet, dann festgestellt, ob nur ganze Pages (Seiten) oder auch ein Restbereich verschoben werden soll. Falls ein solcher Restbereich vorhanden ist, wird auch seine Länge berechnet und zuerst dieser verschoben. Daran schließt sich das Verschieben der ganzen Pages an. Das X- und das Y-Register dienen dabei als Zähler.

Ab $60BB schließt sich wieder unsere eigene Routine an, in der wir die ROMs wieder einschalten. Auf diese Weise lassen sich

Name	BLTUC
Zweck	Verschieben von Speicherinhalten im Speicher
Adresse	$A3BF dez. 41919
Vorbereitungen	Quelle Startadresse nach $5F/60
	Endadresse + 1 nach $5A/5B
	Ziel Endadresse + 1 nach $58/59
Speicherstellen	$58-5B, $5F, $60, $22
Register	Akku, X- und Y-Register
Stapelbedarf	keiner

Bild 9a. BLTUC

noch mehrere Hilfsbildschirme unter ROM-Bereiche packen. Vielleicht überlegen Sie sich mal dazu einen Weg?

Die ROM-Bereiche als Datenquelle

Die ROM-Bereiche enthalten nicht nur ausgeklügelte Maschinenprogramme, sondern auch eine Menge Daten. Sollten Sie mal in die Verlegenheit kommen, beispielsweise die Zahl Pi im MFLPT-Format verwenden zu müssen, dann erfordert das einen ganz schönen Aufwand an Rechen- und Programmarbeit, oder Sie möchten bestimmte Texte wie beispielsweise eine Fehlermeldung verfügbar halten .... und so weiter. Viele von diesen Daten sind schon in der Firmware enthalten und wir werden im folgenden festhalten, wo sie sich befinden und welches Format man vorfindet. Sehen wir uns zunächst Zahlen an (Tabelle 1). Es existieren noch weitere Zahlentabellen in den ROM-Bereichen, die aber selten von Interesse sind. Ebenso wie Zahlen, findet man auch Texte im ROM als ASCII-Werte abgelegt (Tabelle 2):

```
.
,603F  EA        NOP
,6040  A9 00     LDA #00
,6042  85 5F     STA   5F
,6044  A9 E0     LDA #E0
,6046  85 60     STA   60
,6048  A9 E8     LDA #E8
,604A  85 5A     STA   5A
,604C  85 58     STA   58
,604E  A9 E3     LDA #E3
,6050  85 5B     STA   5B
,6052  A9 07     LDA #07
,6054  85 59     STA   59
,6056  20 77 60  JSR 6077
,6059  A9 E9     LDA #E9
,605B  85 5F     STA   5F
,605D  A9 E3     LDA #E3
,605F  85 60     STA   60
,6061  A9 D1     LDA #D1
,6063  85 5A     STA   5A
,6065  A9 E7     LDA #E7
,6067  85 5B     STA   5B
,6069  A9 E8     LDA #E8
,606B  85 58     STA   58
,606D  A9 DB     LDA #DB
,606F  85 59     STA   59
,6071  20 77 60  JSR 6077
,6074  60        RTS
-------------------------------
,6075  EA        NOP
,6076  EA        NOP
,6077  78        SEI
,6078  A5 01     LDA   01
,607A  48        PHA
,607B  A9 35     LDA #35
,607D  85 01     STA   01
,607F  38        SEC
,6080  A5 5A     LDA   5A
,6082  E5 5F     SBC   5F
,6084  85 22     STA   22
,6086  A8        TAY
,6087  A5 5B     LDA   5B
,6089  E5 60     SBC   60
,608B  AA        TAX
,608C  E8        INX
,608D  98        TYA
,608E  F0 23     BEQ 60B3
,6090  A5 5A     LDA   5A
,6092  38        SEC
,6093  E5 22     SBC   22
,6095  85 5A     STA   5A
,6097  B0 03     BCS 609C
,6099  C6 5B     DEC   5B
,609B  38        SEC
,609C  A5 58     LDA   58
,609E  E5 22     SBC   22
,60A0  85 58     STA   58
,60A2  B0 08     BCS 60AC
,60A4  C6 59     DEC   59
,60A6  90 04     BCC 60AC
,60A8  B1 5A     LDA  (5A),Y
,60AA  91 58     STA  (58),Y
,60AC  88        DEY
,60AD  D0 F9     BNE 60A8
,60AF  B1 5A     LDA  (5A),Y
,60B1  91 58     STA  (58),Y
,60B3  C6 58     DEC   58
,60B5  C6 59     DEC   59
,60B7  CA        DEX
,60B8  D0 F2     BNE 60AC
,60BA  68        PLA
,60BB  85 01     STA   01
,60BD  58        CLI
,60BE  60        RTS
-------------------------------
.?
```

Programm 4. Zweiter Teil der Verschieberoutine

```
1 REM ***********************************   <250>
2 REM *                                 *   <229>
3 REM *           PROGRAMM  2           *   <125>
4 REM *                                 *   <231>
5 REM *    ERSTELLEN UND AUFRUF EINES   *   <186>
6 REM *         HILFSBILDSCHIRMES       *   <216>
7 REM *                                 *   <234>
8 REM * HEIMO PONNATH     HAMBURG 1985  *   <082>
9 REM ***********************************   <002>
10 PRINT CHR$(147):POKE 785,0:POKE 786,96:
   GOTO 30                                  <095>
15 REM --------- UP CURSOR SETZEN ------    <112>
20 POKE 211,SP:POKE 214,Z:SYS 58640:RETURN <163>
25 REM- ERSTELLEN DES HILFSBILDSCHIRMES-    <123>
30 Z=1:SP=1:GOSUB 20:PRINT"***************
   ************************"                <151>
40 Z=21:SP=1:GOSUB 20:PRINT"**************
   ************************"                <211>
50 Z=10:SP=7:GOSUB 20:PRINT"TEST FUER DIE
   VERSCHIEBUNG"                            <110>
55 REM ---- AUFRUF ZUM VERSCHIEBEN ----     <033>
60 A=USR(DUMMY)                             <195>
65 REM ---BILDSCHIRM NEU BESCHREIBEN---     <193>
70 GET A$:IF A$=""THEN 70                   <122>
80 PRINT CHR$(147):Z=2:SP=2:GOSUB 20:PRINT
   "JETZT SOLLTE DER ALTE BILDSCHIRM"       <092>
90 Z=4:SP=2:GOSUB 20:PRINT"UNTER DAS KERNA
   L-ROM GESCHOBEN SEIN"                    <150>
100 PRINT:PRINT:PRINT" -- JEDER{2SPACE}USR
   -AUFRUF HOLT DEN --"                     <003>
110 PRINT" -- HILFSBILDSCHIRM WIEDER .{3SP
   ACE}--"                                  <068>
120 PRINT" -- AUCH IM DIREKT-MODUS{7SPACE}
   --"                                      <056>
130 PRINT:PRINT:PRINT"{2SPACE}PROBIEREN SI
   E MAL: A=USR(1) [RETURN]"                <050>
140 Z=19:SP=0:GOSUB 20:END                  <164>
```

© 64'er

Programm 2. Das Demo-Programm zur neuen Verschieberoutine. Vorher müssen Programm 3 und Programm 4 geladen werden.

Startadresse	Format	Zahl
$AEA8	MFLPT	Pi
$B1A5	MFLPT	—32768
$B9BC	MFLPT	1
$B9C1	1-Byte-Integer	3
$B9C2	MFLPT	0.434255942
$B9C7	MFLPT	0.576584541
$B9CC	MFLPT	0.961800759
$B9D1	MFLPT	2.88539007
$B9D6	MFLPT	0.707106781 = SQR(1/2)
$B9DB	MFLPT	1.41421356 = SQR(2)
$B9E0	MFLPT	—0.5
$B9E5	MFLPT	0.693147181 = ln2
$BAF9	MFLPT	10
$BDB3	MFLPT	99999999.9
$BDB8	MFLPT	999999999
$BDBD	MFLPT	1000000000
$BF11	MFLPT	0.5
$BF16	4-Byte-Integer	—100000000
$BF1A	— " —	10000000
$BF1E	— " —	—1000000
$BF22	— " —	100000
$BF26	— " —	—10000
$BF2A	— " —	1000
$BF2E	— " —	—100
$BF32	— " —	10
$BF36	— " —	—1
$BF3A	— " —	—2160000
$BF3E	— " —	216000
$BF42	— " —	—36000
$BF46	— " —	3600
$BF4A	— " —	—600
$BF4E	— " —	60
$BFBF	MFLPT	1.44269504 = 1/ln2
$BFC4	1-Byte-Integer	7
$BFC5	MFLPT	2.14987637E-05
$BFCA	MFLPT	1.43523140E-04
$BFCF	MFLPT	1.34226348E-03
$BFD4	MFLPT	9.61401701E-03
$BFD9	MFLPT	0.0555051269
$BFDE	MFLPT	0.240226385
$BFE3	MFLPT	0.693147186 = ln2
$BFE8	MFLPT	1
$E08D	MFLPT	11879546
$E092	MFLPT	3.92767774E-08
$E2E0	MFLPT	1.57079633 = Pi/2
$E2E5	MFLPT	6.28318531 = 2*Pi
$E2EA	MFLPT	0.25
$E2EF	1-Byte-Integer	5
$E2F0	MFLPT	—14.3813907
$E2F5	MFLPT	42.0077971
$E2FA	MFLPT	—76.7041703
$E2FF	MFLPT	81.6052237
$E304	MFLPT	—41.3417021
$E309	MFLPT	6.28318531 = 2*Pi
$E33E	1-Byte-Integer	11
$E33F	MFLPT	—6.8473912E—04
$E344	MFLPT	4.85094216E—03
$E349	MFLPT	—0.0161117018
$E34E	MFLPT	0.034209638
$E353	MFLPT	—0.0542791328
$E358	MFLPT	0.0724571965
$E35D	MFLPT	—0.0898023954
$E362	MFLPT	0.110932413
$E367	MFLPT	—0.142839808
$E36C	MFLPT	0.19999912
$E371	MFLPT	—0.333333316
$E376	MFLPT	1
$E3BA	MFLPT	0.811635157
$E8DA - $E8E9	1-Byte-Integers	Tabelle der Farbcodes
$EB81 - $EBC1	— " —	Tastaturdecodierung: Einzelne Tasten
$EBC2 - $EC02	1-Byte-Integers	Tasten mit Shift
$EC03 - $EC43	1-Byte-Integers	Tasten mit Commodore-Taste
$EC78 - $ECB8	1-Byte Integers	Tasten mit Control-Taste
$ECB9 - $ECE5	1-Byte-Integers	VIC-II-Chip-Registerwerte
$ECF0 - $ED08	1-Byte-Integers	Tabelle der LSBs der Bildschirm-Anfangsadressen

Tabelle 1. Im ROM stehen nicht nur Programme, sondern auch Tabellen, hier einige wichtige Zahlen.

```
.
,5000  A2 00     LDX #00
,5002  86 5C     STX   5C
,5004  86 5D     STX   5D
,5006  A0 10     LDY #10
,5008  06 57     ASL   57
,500A  26 58     ROL   58
,500C  26 5C     ROL   5C
,500E  26 5D     ROL   5D
,5010  38        SEC
,5011  A5 5C     LDA   5C
,5013  E5 59     SBC   59
,5015  AA        TAX
,5016  A5 5D     LDA   5D
,5018  E5 5A     SBC   5A
,501A  90 06     BCC 5022
,501C  86 5C     STX   5C
,501E  85 5D     STA   5D
,5020  E6 57     INC   57
,5022  88        DEY
,5023  D0 E3     BNE 5008
,5025  60        RTS
-----------------------------------
.?
```

Programm 1. Die 16-Bit-Division

```
.
,6000  A9 00     LDA #00
,6002  85 5F     STA   5F
,6004  A9 04     LDA #04
,6006  85 60     STA   60
,6008  A9 E8     LDA #E8
,600A  85 5A     STA   5A
,600C  85 58     STA   58
,600E  A9 07     LDA #07
,6010  85 5B     STA   5B
,6012  A9 E3     LDA #E3
,6014  85 59     STA   59
,6016  20 BF A3  JSR A3BF
,6019  A9 00     LDA #00
,601B  85 5F     STA   5F
,601D  A9 D8     LDA #D8
,601F  85 60     STA   60
,6021  A9 E8     LDA #E8
,6023  85 5A     STA   5A
,6025  A9 DB     LDA #DB
,6027  85 5B     STA   5B
,6029  A9 D1     LDA #D1
,602B  85 58     STA   58
,602D  A9 E7     LDA #E7
,602F  85 59     STA   59
,6031  20 BF A3  JSR A3BF
,6034  A9 40     LDA #40
,6036  8D 11 03  STA 0311
,6039  A9 60     LDA #60
,603B  8D 12 03  STA 0312
,603E  60        RTS
-----------------------------------
.?
```

Programm 3. Zweiter Teil der Verschieberoutine

$A004	CBMBASIC
$A09E - $A19D	Texte der Basic-Befehlsworte (im letzten Byte ist jeweils Bit 7 gesetzt)
$A19E - $A327	Texte der Basic-Fehler- und System-Meldungen. (Im letzten Byte ist jeweils Bit 7 gesetzt)
$A364 - $A38A	Weitere System-Meldungen: OK, ERROR, IN, READY, BREAK. (Das letzte Byte ist jeweils 0)
$ACFC - $AD1D	Fehlermeldungstexte für INPUT: ?EXTRA IGNORED, ?REDO FROM START. (Das letzte Byte ist jeweils 0)
$E460	BASIC BYTES FREE
$E473	**** COMMODORE 64 BASIC V2 **** 64K-RAM-System
$ECE6	LOAD (Return) RUN (Return)
$F0BD - $F12B	Texte für Ein- und Ausgabe-Operationen
$FD10	CBM80

Tabelle 2. Diese Texte sind im ROM als ASCII-Werte abgelegt

Befehlswort	Adressierung	Bytezahl	Code Hex	Code Dez	Taktzyklen	Beeinflussg. von Flaggen
LSR	»Akkumulator«	1	1A	26	2	N,Z,C
	absolut	3	4E	78	6	N,Z,C
	0-page-absolut	2	46	70	5	N,Z,C
	absolut-X-indiz.	3	5E	94	7	N,Z,C
	0-page-X-indiz.	2	56	86	6	N,Z,C
ROL	»Akkumulator«	1	2A	42	2	N,Z,C
	absolut	3	2E	46	6	N,Z,C
	0-page-absolut	2	26	38	5	N,Z,C
	absolut-X-indiz.	3	3E	62	7	N,Z,C
	0-page-X-indiz.	2	36	54	6	N,Z,C
ROR	»Akkumulator«	1	6A	106	2	N,Z,C
	absolut	3	6E	110	6	N,Z,C
	0-page-absolut	2	66	102	5	N,Z,C
	absolut-X-indiz.	3	7E	126	7	N,Z,C
	0-page-X-indiz.	2	76	118	6	N,Z,C

Tabelle 3. Die in dieser Ausgabe besprochenen Assembler-Befehle

Sollten Sie mal in die Verlegenheit kommen, solche Texte ausgeben zu wollen, dann legen Sie sie nicht nochmal in einer eigenen Texttabelle ab, sondern schöpfen Sie aus dem Fundus, den wir im ROM-Bereich fix und fertig haben.

Diese Folge soll nicht abgeschlossen werden, ohne eine Korrektur. Auf einen Fehler, dem ich aufgesessen bin (in der Literatur befinde ich mich aber in guter Gesellschaft, andere sind auch davon betroffen), haben mich zwei aufmerksame Leser hingewiesen. Es dreht sich um die Flaggensetzung bei Compare-Befehlen. Die N-Flagge ist nämlich nicht nur vom Ergebnis des Vergleichs, sondern auch noch von den aktuellen Akku- beziehungsweise Registerinhalten bestimmt.

Bild 1 in der 5. Folge muß deshalb korrigiert werden:
(A,X,Y) größer als die Daten: N kann 0 oder 1 sein
(A,X,Y) = Daten N = 0
(A,X,Y) kleiner als die Daten: N kann 0 oder 1 sein.

Das stammt aus dem offziellen MOS-Technology-Handbuch und entspricht somit hoffentlich der Wahrheit [2]. Das bedeutet, daß man bei den Abfragen durch Branch-Befehle nach den Vergleichsbefehlen etwas vorsichtig sein sollte, was die N-Flagge angeht.

Zum Schluß noch, wie üblich, die Tabelle 3 mit den neuen Assembler-Befehlen.

(Heimo Ponnath/gk)

[1] »Computerspiele und Wissenswertes Commodore 64«, Haar bei München: Markt & Technik Verlag, 1984. Das ist die von P. Lücke besorgte Übersetzung des amerikanischen Buches »More on the sixtyfour« und ist jedem Assembler-Programmierer warm zu empfehlen.
[2] »MOS Microcomputers Programmier-Handbuch«, Frankfurt: Commodore MOS Technology

Tips & Tricks gesucht



Jeder Computer und jedes Programm hat seine speziellen Schwachstellen und Unzulänglichkeiten. Allerdings ist kaum ein Programmierer oder Anwender auf Dauer bereit, sich damit abzufinden. Wo auch sorgfältigste Lektüre von Handbüchern nicht weiterhilft, da wird so manche Stunde experimentiert, um eine Lösung zu finden (die oft in einer Basic-Zeile Platz hat).

Wir suchen solche Tips und Tricks, um sie allen Lesern zugänglich zu machen. Schließlich ist es wenig sinnvoll, sich wochenlang mit Problemen herumzuschlagen, die andere bereits gelöst haben.

Wenn Sie also interessante Tips für den Umgang mit Computer, Floppy, Drucker oder sonstiger Hardware haben, wenn Sie bei kommerzieller Software einige Kniffe kennen, die nicht in der Anleitung stehen, oder wenn Sie interessante Problemlösungen statt in vier Seiten Listing in ein oder zwei Basic-Zeilen untergebracht haben, dann sollten Sie uns auf jeden Fall einmal schreiben.

Bitte geben Sie genau den Computertyp und die Gerätekonfiguration oder die Software an, und senden Sie Ihren Tip oder Trick an die

Redaktion 64'er
Markt & Technik Verlag
Aktiengesellschaft
Hans-Pinsel-Str. 2
8013 Haar bei München

In die Geheimnisse der Floppy eingetaucht

Formatieren einer Diskette ist für jeden Floppy-Besitzer das erste, was er mit ihr macht. Was beim Formatieren passiert und weshalb die Floppy so nervig rattert, warum es so lange dauert und wie es schneller geht, erfahren Sie in folgendem Artikel.

Teil 6

Wie jedem Floppy-Besitzer bekannt ist, muß eine Diskette vor dem ersten Speichern von Daten formatiert werden. Wie eine Diskette nach einem solchen Formatiervorgang aussieht, wurde schon besprochen.

Uns soll nun interessieren, was während des Formatierens so alles in der Floppy passiert und warum die 1541 so lange für einen eigentlich sehr einfachen Vorgang benötigt.

Zur Wiederholung: Beim Formatieren werden vom DOS alle wichtigen Markierungen auf die Diskette gebracht und außerdem sämtliche Sektoren in ihrer späteren Form angelegt.

Der Vorgang des Formatierens verwendet zu seiner Ausführung einen uns schon bekannten Jobcode, nämlich $E0.

Bevor das DOS den eigentlichen Formatiervorgang startet, wird ab $0600 (also im Puffer 3) ein Sprungbefehl abgelegt: JMP $FAC7.

Dieser Sprungbefehl ist eine Art Vektor, der im RAM liegt und somit verändert werden kann. Er bietet dem Benutzer die Möglichkeit, eine eigene Routine einzubauen, die dann bei jedem Trackwechsel angesprungen wird, um so einige wirksame Manipulationen an der Formatierung vorzunehmen, indem zum Beispiel Werte in der Zeropage verändert werden, doch dazu später. Üblicherweise zeigt dieser Vektor direkt auf eine Jobroutine, die für das Formatieren zuständig ist. Diese Routine wird nun vom Hauptprogramm mit dem Jobcode $E0, der in Speicherstelle $03 geschrieben wird, aufgerufen.

Formatieren in der Jobschleife

Am Anfang der Jobroutine steht nun die Abfrage, ob schon mindestens ein Track formatiert wurde oder ob dieser Einsprung der allererste ist. Ist dieser Einsprung der erste, so werden alle Parameter für den Steppermotor gesetzt; danach erfolgt ein Rücksprung in die Jobschleife. Hier wird der Tonkopf nun 45 (!) Tracks zurückgefahren, was sich in jenem charakteristischen Rattern der Floppystation äußert.

Nun, können Sie sagen, es würde auch reichen, wenn der Kopf nur 35 oder 40 Spuren zurücktransportiert würde. In der Tat ist der Wert 45 sehr hoch. Man muß aber bedenken, daß es passieren kann, daß der Schreib-/Lesekopf der Floppy durch eine defekte Diskette oder einen Programmierfehler zu weit nach innen gefahren und beispielsweise auf Track 42 am Anschlag gelandet ist, daß ein Zurückfahren um 40 Tracks einfach nicht ausreicht, um den Tonkopf richtig zu positionieren.

Der Wert von 45 Tracks enthält also eine Sicherheitsreserve, die ein Positionieren auf Spur 1 mit Sicherheit ermöglicht.

Wurde der Kopf also auf Track 1 positioniert, so erfolgt erneut ein Einsprung in die Formatierungsroutine; eine Speicherstelle zeigt jetzt an, daß der Kopf auf Track 1 positioniert wurde und das Formatieren starten kann.

Jetzt wird noch geprüft, ob auf die nächste Spur umgeschaltet werden soll, da die aktuelle bereits formatiert wurde (wenn ja, erfolgt wieder ein Einsprung in die Jobschleife, um das Nötige zu tun).

Diese Abfragen am Anfang der Formatierungsroutine scheinen umständlich und überflüssig zu sein; das Gefühl täuscht jedoch. Wir dürfen ja nicht vergessen, daß die Routine immer nur jeweils einen Track formatiert und danach zur Jobschleife zurückkehrt, damit der Tonkopf weitergeführt werden kann. Wir haben also gewissermaßen eine Endlosschleife, die nur durch die Feststellung, daß Spur 35 fertig formatiert wurde, beendet wird.

Ausmessen einer Spur

Jetzt haben wir aber endlich alle Voraussetzungen zum Formatieren eines Tracks erfüllt und wollen an die Arbeit gehen. Der Abschnitt, der jetzt besprochen wird, ist übrigens für die langwierige Formatierung verantwortlich und sorgt für die ausgedehnten Wartezeiten.

Bevor die SYNC-Markierungen und Sektoren auf eine Spur geschrieben werden, wird diese Spur vom DOS »ausgemessen«.

Das Betriebssystem der 1541 »weiß« im Normalfall genau, wieviele Bytes für die SYNC-Markierungen und Sektoren einer Spur benötigt, beziehungsweise verbraucht werden.

Jetzt ist es aber so, daß die Sektoren nicht genau auf jede Spur abgemessen sind; vielmehr hat die Diskette pro Spur eine etwas höhere Kapazität, als eigentlich benötigt wird. Aus dieser Tatsache folgt natürlich, daß zwischen den einzelnen Sektoren »Leerstellen« entstehen, die keine Daten enthalten.

Da jetzt aber die Länge der Tracks von außen (Track 1) nach innen (Track 35) kontinuierlich abnimmt, werden diese Leerstellen immer kleiner; wir haben also unterschiedliche Anzahlen von »Leerbytes« zwischen den Sektoren.

Das DOS ist nun bestrebt, die Sektoren jeder Spur möglichst symmetrisch anzuordnen, also immer den gleichen Abstand zwischen zwei Sektoren eines Tracks zu haben. Bild 1 zeigt, was passiert, wenn keine vorherige Ausmessung stattfindet.

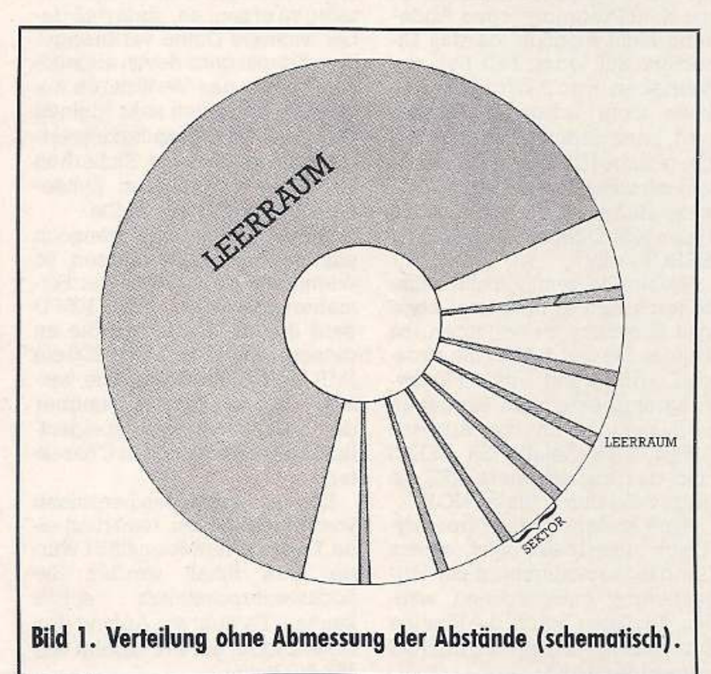


Bild 1. Verteilung ohne Abmessung der Abstände (schematisch).

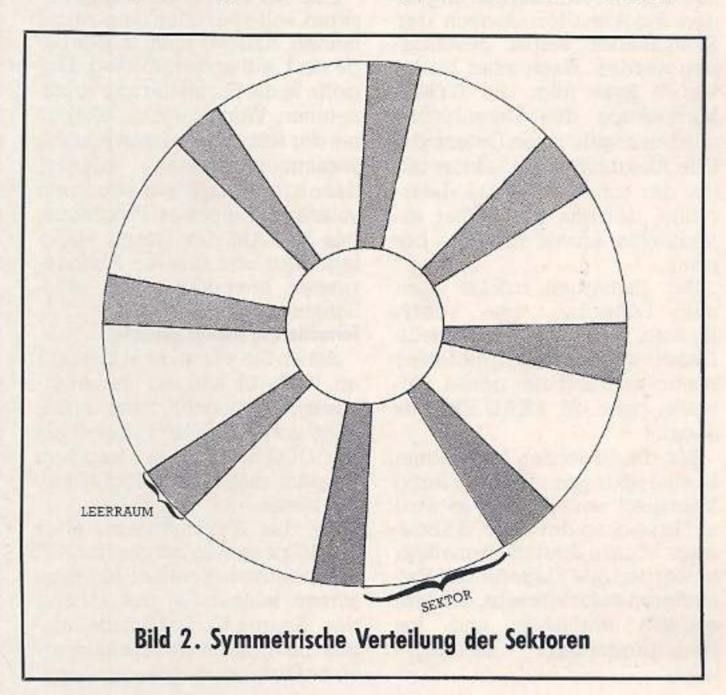


Bild 2. Symmetrische Verteilung der Sektoren

Um das Ziel einer »symmetrisch« formatierten Diskette zu erreichen, stellt das DOS durch einige komplizierte Schreib- und Lesevorgänge das Verhältnis zwischen benötigtem und vorhandenem Platz einer Spur fest. Aus diesem Verhältnis kann nun anhand einer einfachen Rechnung festgestellt werden, wieviel Platz zwischen den einzelnen Sektoren freigelassen werden muß.

Nachdem diese komplizierte Vermessung stattgefunden hat, die mehrere Diskettenumdrehungen und damit Zeit erfordert, beginnt nun das eigentliche Formatieren der Diskette, das mit allem Drum und Dran normalerweise nicht mehr als eine ⅓ Sekunde für eine Track benötigt.

Das Anlegen der Sektoren im Puffer

Bevor geschrieben werden kann, müssen die Sektoren erst einmal im Pufferspeicher der 1541 hergestellt werden. Da sich die einzelnen Sektoren nur durch deren Header unterscheiden, reicht das Anlegen der Blockheader; die Inhalte der Datenblöcke sind bei jedem Sektor gleich und bestehen aus dem schon bekannten Muster $4B gefolgt von 255 $01-Bytes.

Die Blockheader werden alle in einem Pufferspeicher ($0300-$03FF) abgelegt; der Inhalt der Datenblöcke steht ab $0500 bis $05FF.

Schreiben eines Tracks auf Diskette

So, alle Vorarbeiten wären jetzt abgeschlossen. Wir können mit dem Schreiben auf Diskette beginnen. Zuerst wird der Disk-Controller auf Schreibmodus gestellt und die Spur der Diskette gelöscht.

Der gesamte Spurinhalt wird nun während einer einzigen Diskettenumdrehung (⅕ Sekunde) auf die Diskette gebracht, wobei zuerst die SYNC-Markierung für den Blockheader, danach der Blockheader selbst geschrieben werden. Nach einer Lücke von 9 Byte folgt die SYNC-Markierung des Datenblocks mit den zugehörigen Datenbyte. Den Abschluß eines Sektors bildet der schon erwähnte »Leerraum«, der aus der vorher errechneten Anzahl von Byte besteht.

Zur Sicherheit erfolgt nach dem Schreiben eine Verify-Routine, die auf eventuelle Disketten- oder Schreibfehler kontrolliert und bei deren Auftreten einen »24, READ ERROR« ausgibt.

Mit dieser letzten Maßnahme ist eine Spur einer Diskette fertig formatiert worden, und es wird auf Erreichen der Spur 35 abgefragt. Wurde Spur 35 formatiert, so werden alle Flags für das Formatieren zurückgesetzt, die Jobschleife verlassen und ins Hauptprogramm zurückgekehrt.

Formatieren ohne ID

Im Hauptprogramm wird nun auf Track 18 positioniert. Die BAM der Diskette wird hergestellt und in Block 18,0 abgelegt. Anschließend wird noch der erste Directory-Block (18,1) mit Nullen vollgefüllt und ebenfalls abgespeichert, womit das Formatieren abgeschlossen wäre.

Formatiert man eine Diskette nur kurz, das heißt ohne Angabe einer ID beim N-Befehl, so werden alle anfänglichen Schritte weggelassen. Es wird in diesem Fall nur auf das richtige Formatkennzeichen in der BAM ($41/65/A) kontrolliert und danach der eben beschriebene Vorgang auf Track 18 durchgeführt.

Formatieren mit »Variationen«

Nun wäre unser Floppykurs natürlich kein Floppykurs, wenn wir unsere neu gewonnenen theoretischen Kenntnisse nicht sofort in die Praxis umsetzen wollten.

In der Tat kann man mit Hilfe der Formatierroutine im DOS einige nette »Scherze« auf eine Diskette bringen, die entweder dem Spieltrieb oder dem Softwareschutz dienen können.

Ich habe vorhin schon erwähnt, daß die Formatierroutine jeweils über einen Sprungbefehl bei $0600 im RAM der Floppy aufgerufen wird.

Diese Adresse wird bei jedem neuen Track angesprungen und bietet so die Möglichkeit, Tracks zu erzeugen, die in ihrem Aufbau voneinander abweichen, wenn entsprechende Eingriffe vorgenommen werden.

Diese Möglichkeit eines Eingriffes wollen wir an dieser Stelle aber gar nicht erst weiter diskutieren, da es ziemlich aussichtslos ist, hier ohne dokumentiertes DOS-Listing an die Arbeit zu gehen.

Daß wir kein DOS Listing besitzen, soll aber noch lange nicht heißen, daß wir nicht in der Lage sind, auf anderem Weg, Eingriffe in die Formatierung vorzunehmen. Wenn wir nicht effektiv mit der fest eingebauten Routine zusammenarbeiten können, dann schreiben wir uns eben ein vollständig eigenes Programm, das im RAM der Floppy abgelegt wird und uns für Abänderungen unendlich viele Möglichkeiten bietet.

Formatierung »selbst gebaut«

Sehen Sie sich einmal Listing 1 an. Ich habe hier ein Formatiersystem entwickelt, das einfacher und schneller arbeitet als die DOS-Routine und trotzdem ein paar zusätzliche Möglichkeiten bietet.

Da das Gesamtprinzip aber fast 100prozentig mit der im DOS eingebauten Routine übereinstimmt, können Sie sich anhand des Source-Code-Listings einmal die »praktische Ausführung« einer Formatierroutine ansehen.

Um Ihnen die Eingabe des Programms zu erleichtern, habe ich einen DATA-Lader als Listing 2 beigefügt, wobei ich Ihnen empfehlen möchte, diesen gleich einmal einzutippen.

Das Programm wird nur aktiviert, wenn alle DATAs richtig eingetippt wurden. Haben Sie alles richtig gemacht, so steht nach der Ausführung des Laders ein Maschinenprogramm am Basic-Anfang, dem eine Basic-Zeile beigefügt ist. Das Programm sollten Sie sich jetzt mit SAVE auf eine Diskette speichern und danach mit RUN starten.

Nach einer winzigen Verzögerung erscheint die READY-Meldung und der Cursor wieder. Das Formatierungsprogramm wurde jetzt in den Bereich ab $C000 (49152) geschoben und der SAVE-Vektor abgeändert.

Tippen Sie jetzt einfach den Befehl SAVE — ohne Anführungszeichen und Filenamen — ein und drücken Sie RETURN. Es erscheint nun die Startmeldung des Formatprogrammes. Sie können jetzt einen Namen für eine Diskette eingeben (maximal 16 Zeichen werden angenommen). Danach erwartet der Computer eine zweistellige ID. Schließlich, und das ist das besondere an diesem Programm, können Sie noch den ersten und letzten zu formatierenden Track eingeben. Diese Eingabe muß hexadezimal erfolgen und erlaubt einen Bereich von $01 bis $FF.

Achtung! Wird eine Zahl größer als $29 (41) eingegeben, wird es in der Regel kritisch. Der Kopf ist dann nämlich am oberen Anschlagpunkt angelangt.

Etwas ist noch zu beachten: Ein Nachformatieren einer Spur auf einer gefüllten Diskette ist mit dem Programm ohne Änderung nicht möglich, da das Directory auf jeden Fall neu geschrieben wird. Wird die Diskette nicht vollständig formatiert, so ist darauf zu achten, daß die gleiche ID eingegeben wird, wie sie schon für die übrige Diskette Gültigkeit hat, da es sonst einen »29, DISK ID MISMATCH ERROR« gibt.

Wollen Sie dennoch einen Einzeltrack neu formatieren, ohne das Directory zu zerstören, so können Sie das durch eine einfache Änderung im Floppy-Programm erreichen. Sie gehen in Listing 1a an die Adresse $06B5. Den Befehl JSR $EE40 und das nachfolgende RTS ersetzen Sie durch lauter NOPs.

Eine Änderung des Directory-Track unterbleibt jetzt, sofern Sie die Tracknummern zur Formatierung entsprechend wählen, da dieser Befehl die Routine zum kurzen Formatieren im DOS aufgerufen hätte.

In jedem Fall gilt aber: Bei Formatieren von Einzeltracks müssen diese die gleiche ID wie die übrige Diskette erhalten.

Eine weitere Möglichkeit dient der Schonung des Laufwerks. Wenn Sie sich das Floppy-Programm noch einmal betrachten, dann finden Sie bei Adresse $0696 den Befehl an den Disk-Controller, einen BUMP auszuführen. Wenn Sie hier das $C0 durch ein $00 ersetzen, dann unterbleibt dieses Anschlagen des Tonkopfes am Anfang des Formatierens. Diese Maßnahme ist immer dann nützlich, wenn mehrere Disketten hintereinander formatiert werden sollen.

Zur Zeitdauer ist noch zu sagen, daß das Programm für eine Diskette zirka 30 Sekunden benötigt und damit um einiges schneller ist als das Programm im DOS der 1541. Warum das so ist, wollen wir gleich erfahren.

Geschwindigkeit; aber wie?

In meinem Formatierprogramm wurde die Berechnung der Lücke zwischen zwei Sektoren weggelassen. Wir können nämlich davon ausgehen, daß diese Lücken auf jeder Diskette in etwa gleich sind. Aus diesem Grund verwende ich einfach einen Erfahrungswert für die Länge der Lücke, der zusätzlich noch einen Sicherheitsbereich enthält. Diesen Wert sehen Sie in Listing 1a an der Adresse $05DF.

Wenn Sie mit dem Programm Disketten formatieren, werden Sie feststellen, daß die Datensicherheit auch weiterhin voll gewährleistet ist.

Im Gegensatz zu anderen schnellen Formatierprogrammen habe ich aber nicht auf ein Verify verzichtet, da das Formatieren die einzige Möglichkeit bietet, defekte Disketten rechtzeitig zu erkennen, ohne daß dabei wichtige Daten verlorengehen. Einmal ganz davon abgesehen, macht das Verifizieren außerdem nur einen sehr kleinen Teil am Geschwindigkeitsverlust aus, so daß die Sicherheit vor einigen Sekunden Zeitgewinn Vorrang haben sollte.

Wollen Sie die Zeit dennoch einmal ohne Verify messen, so »klemmen« sie den Rest der Formatierungsroutine ab $05FD ganz einfach ab, indem Sie an dieser Stelle nach JMP $FE00 ein JMP $FD9E einfügen. Eine weitere Verbesserung gegenüber dem DOS 2.6 der 1541 hat eigentlich mehr kosmetischen Charakter.

Es geht hier um den Leerinhalt von Datenblöcken, nachdem eine Diskette neu formatiert wurde. Den Inhalt werden Sie höchstwahrscheinlich schon kennen: Es steht am Anfang des Datenblocks ein $4B gefolgt von 255 $01-Bytes. ►

```
Floppyprogramm zum Disk-Format-System
    1985 by KOSS
0500  ea         nop
0501  a5 0a      lda $0a       Tracknummer aus Jobspeicher
0503  c9 24      cmp #$24      größer als 35?
0505  90 07      bcc $050e     nein
0507  a9 12      lda #$12      ja: 18 als Anzahl der Sektoren
0509  85 43      sta $43       festlegen
050b  4c 13 05   jmp $0513
050e  20 4b f2   jsr $f24b     Anzahl der Sektoren holen
0511  85 43      sta $43       und merken
0513  a9 00      lda #$00
0515  85 1b      sta $1b       Sektorzähler setzen
0517  a0 00      ldy #$00
0519  a2 00      ldx #$00
051b  a5 39      lda $39       Kennzeichen $08 für Blockheader
051d  99 00 03   sta $0300,y
0520  c8         iny
0521  c8         iny
0522  a5 1b      lda $1b       Sektornummer
0524  99 00 03   sta $0300,y
0527  c8         iny
0528  a5 0a      lda $0a       Tracknummer
052a  99 00 03   sta $0300,y
052d  c8         iny
052e  a5 13      lda $13       ID 2
0530  99 00 03   sta $0300,y
0533  c8         iny
0534  a5 12      lda $12       ID 1
0536  99 00 03   sta $0300,y
0539  c8         iny
053a  a9 0f      lda #$0f
053c  99 00 03   sta $0300,y   Lücke lassen
053f  c8         iny
0540  99 00 03   sta $0300,y
0543  c8         iny
0544  a9 00      lda #$00
0546  59 fa 02   eor $02fa,y   Prüfsumme bilden
0549  59 fb 02   eor $02fb,y
054c  59 fc 02   eor $02fc,y
054f  59 fd 02   eor $02fd,y
0552  99 f9 02   sta $02f9,y   und abspeichern
0555  e6 1b      inc $1b
0557  a5 1b      lda $1b       Sektorzähler erhöhen
0559  c5 43      cmp $43       schon Maximalzahl?
055b  90 be      bcc $051b     nein, weitermachen
055d  a9 03      lda #$03
055f  85 31      sta $31
0561  98         tya
0562  48         pha
0563  8a         txa
0564  9d 00 07   sta $0700,x   Datenblock mit $00 füllen
0567  e8         inx
0568  d0 fa      bne $0564
056a  20 30 fe   jsr $fe30
056d  68         pla
056e  a8         tay
056f  88         dey
0570  20 e5 fd   jsr $fde5
0573  20 f5 fd   jsr $fdf5
0576  a9 07      lda #$07
0578  85 31      sta $31
057a  20 e9 f5   jsr $f5e9     Prüfsumme für Datenblock
057d  85 3a      sta $3a       abspeichern
057f  20 8f f7   jsr $f78f
0582  a9 00      lda #$00      Sektorzähler setzen
0584  85 32      sta $32
0586  20 0e fe   jsr $fe0e     Track löschen
0589  a9 ff      lda #$ff
058b  8d 01 1c   sta $1c01     SYNC schreiben
058e  a2 05      ldx #$05
0590  50 fe      bvc $0590
0592  b8         clv
0593  ca         dex
0594  d0 fa      bne $0590
0596  a2 0a      ldx #$0a
0598  a4 32      ldy $32
059a  50 fe      bvc $059a     Blockheader schreiben
059c  b8         clv
059d  b9 00 03   lda $0300,y
05a0  8d 01 1c   sta $1c01
05a3  c8         iny
05a4  ca         dex
05a5  d0 f3      bne $059a
05a7  a2 09      ldx #$09
05a9  50 fe      bvc $05a9     Lücke von 9 Bytes lassen
05ab  b8         clv
05ac  a9 55      lda #$55
05ae  8d 01 1c   sta $1c01
05b1  ca         dex
05b2  d0 f5      bne $05a9
05b4  a9 ff      lda #$ff
05b6  a2 05      ldx #$05
05b8  50 fe      bvc $05b8     SYNC-Markierung für Datenblock
05ba  b8         clv
05bb  8d 01 1c   sta $1c01
05be  ca         dex
05bf  d0 f7      bne $05b8
05c1  a2 bb      ldx #$bb
05c3  50 fe      bvc $05c3
05c5  b8         clv
05c6  bd 00 01   lda $0100,x
05c9  8d 01 1c   sta $1c01
05cc  e8         inx
05cd  d0 f4      bne $05c3
05cf  a0 00      ldy #$00
05d1  50 fe      bvc $05d1     Datenblock schreiben
05d3  b8         clv
05d4  b1 30      lda ($30),y
05d6  8d 01 1c   sta $1c01
05d9  c8         iny
05da  d0 f5      bne $05d1
05dc  a9 55      lda #$55
05de  a2 08      ldx #$08      Lücke nach Sektor mit fester
05e0  50 fe      bvc $05e0     Länge von 8 Bytes schreiben
05e2  b8         clv
05e3  8d 01 1c   sta $1c01
05e6  ca         dex
05e7  d0 f7      bne $05e0
05e9  a5 32      lda $32
05eb  18         clc
05ec  69 0a      adc #$0a
05ee  85 32      sta $32
05f0  c6 1b      dec $1b       schon alle Sektoren?
05f2  d0 95      bne $0589     nein, weitermachen
05f4  50 fe      bvc $05f4
05f6  b8         clv
05f7  50 fe      bvc $05f7
05f9  b8         clv
05fa  20 00 fe   jsr $fe00     auf Lesen umschalten
05fd  a9 c8      lda #$c8      200 Leseversuche
05ff  85 1f      sta $1f
0601  a9 00      lda #$00
0603  85 30      sta $30
0605  a9 03      lda #$03
0607  85 31      sta $31
0609  a5 43      lda $43
060b  85 1b      sta $1b       Sektorzähler
060d  20 56 f5   jsr $f556     auf SYNC-Signal warten
0610  a2 0a      ldx #$0a
0612  a0 00      ldy #$00
0614  50 fe      bvc $0614
0616  b8         clv
0617  ad 01 1c   lda $1c01
061a  d1 30      cmp ($30),y   Daten vergleichen
061c  d0 0e      bne $062c
061e  c8         iny
061f  ca         dex
0620  d0 f2      bne $0614
0622  18         clc
0623  a5 30      lda $30
0625  69 0a      adc #$0a
0627  85 30      sta $30
0629  4c 35 06   jmp $0635
062c  c6 1f      dec $1f       Zähler vermindern
062e  d0 d1      bne $0601
0630  a9 06      lda #$06
0632  4c d3 fd   jmp $fdd3     24, READ ERROR
0635  20 56 f5   jsr $f556     SYNC-Signal abwarten
0638  a0 bb      ldy #$bb
063a  50 fe      bvc $063a
063c  b8         clv
063d  ad 01 1c   lda $1c01
0640  d9 00 01   cmp $0100,y
0643  d0 e7      bne $062c
0645  c8         iny
0646  d0 f2      bne $063a
0648  a2 fc      ldx #$fc
064a  50 fe      bvc $064a
064c  b8         clv
064d  ad 01 1c   lda $1c01
0650  d9 00 07   cmp $0700,y   Datenblock testen
0653  d0 d7      bne $062c
0655  c8         iny
0656  ca         dex
0657  d0 f1      bne $064a
0659  c6 1b      dec $1b
065b  d0 b0      bne $060d
065d  4c 9e fd   jmp $fd9e     Ende; zur Jobschleife

0660  a0 00      ldy #$00      Start des Floppyprogramms
0662  b9 e0 06   lda $06e0,y
0665  99 00 02   sta $0200,y   Disknamen übernehmen
0668  c8         iny
0669  cc df 06   cpy $06df
066c  90 f4      bcc $0662
066e  ad df 06   lda $06df
0671  8d 74 02   sta $0274     Länge der Zeile setzen
0674  ad de 06   lda $06de
0677  8d 7b 02   sta $027b     Kommaposition setzen
067a  a9 00      lda #$00
067c  85 7f      sta $7f       Drive 0 setzen
067e  20 00 c1   jsr $c100     LED am Laufwerk an
0681  ac 7b 02   ldy $027b
0684  b9 00 02   lda $0200,y   ID 1 holen
0687  85 12      sta $12
0689  b9 01 02   lda $0201,y   id 2 holen
068c  85 13      sta $13
068e  20 07 d3   jsr $d307     alle Kanäle schließen
0691  a9 1a      lda #$1a
0693  8d 05 1c   sta $1c05     Timer setzen
0696  a9 c0      lda #$c0      BUMP anfordern
0698  85 00      sta $00
069a  a5 00      lda $00
069c  30 fc      bmi $069a     auf Ausführung warten
069e  ae dc 06   ldx $06dc     erste Tracknummer
06a1  86 0a      stx $0a
06a3  a9 e0      lda #$e0      Track formatieren
06a5  85 02      sta $02
06a7  a5 02      lda $02

06a9  30 fc      bmi $06a7     auf Ende warten
06ab  c9 02      cmp #$02      Fehler aufgetreten?
06ad  b0 0c      bcs $06bb     verzweige, wenn ja
06af  e8         inx
06b0  ec dd 06   cpx $06dd     schon Zieltrack formatiert?
06b3  90 ec      bcc $06a1     weiter, wenn nein
06b5  20 40 ee   jsr $ee40     Directory herstellen
06b8  60         rts           Ende
```

Listing 1a. Das Floppy-Programm zum Format-System

```
06b9  ea         nop
06ba  ea         nop
06bb  a2 02      ldx #$02
06bd  4c 0a e6   jmp $e60a    Diskstatus ausgeben; Ende

   Computerprogramm zum Disk-Format-System
   (c) 1985 by KOSS
., c200 a2 00    ldx #$00
., c202 20 87 c2 jsr $c287          Titel und erste Frage ausgeben
., c205 a0 00    ldy #$00
., c207 20 cf ff jsr $ffcf          Eingabe holen
., c20a c9 0d    cmp #$0d
., c20c f0 08    beq $c216
., c20e 99 e0 c1 sta $c1e0,y        Namen abspeichern
., c211 c8       iny
., c212 c0 10    cpy #$10           schon 16 Zeichen ?
., c214 90 f1    bcc $c207          weiter, wenn nein
., c216 a9 2c    lda #$2c           Komma hinter den Namen setzen
., c218 99 e0 c1 sta $c1e0,y
., c21b c8       iny
., c21c 8c de c1 sty $c1de
., c21f a2 47    ldx #$47
., c221 20 87 c2 jsr $c287          Frage nach Disk-ID
., c224 a2 00    ldx #$00
., c226 20 cf ff jsr $ffcf          Eingabe abwarten
., c229 c9 0d    cmp #$0d
., c22b f0 09    beq $c236
., c22d 99 e0 c1 sta $c1e0,y        ID ebenfalls abspeichern
., c230 c8       iny
., c231 e8       inx
., c232 e0 02    cpx #$02
., c234 90 f0    bcc $c226
., c236 8c df c1 sty $c1df
., c239 a2 53    ldx #$53
., c23b 20 87 c2 jsr $c287          'FROM TRACK:$' ausgeben
., c23e 20 cf ff jsr $ffcf
., c241 85 fa    sta $fa
., c243 20 cf ff jsr $ffcf
., c246 85 fb    sta $fb
., c248 a9 00    lda #$00
., c24a 85 d0    sta $d0
., c24c a2 62    ldx #$62
., c24e 20 87 c2 jsr $c287          'TO TRACK:$' ausgeben
., c251 20 cf ff jsr $ffcf
., c254 85 fc    sta $fc
., c256 20 cf ff jsr $ffcf
., c259 85 fd    sta $fd
., c25b a9 00    lda #$00
., c25d 85 d0    sta $d0
., c25f a5 fa    lda $fa
., c261 a6 fb    ldx $fb
., c263 20 04 c4 jsr $c404          Umrechnung in HEX-Byte
., c266 8d dc c1 sta $c1dc          Anfangstrack setzen
., c269 a5 fc    lda $fc
., c26b a6 fd    ldx $fd
., c26d 20 04 c4 jsr $c404          Umrechnung in HEX-Byte
., c270 8d dd c1 sta $c1dd          Endetrack setzen
., c273 ee dd c1 inc $c1dd          plus 1 als Vergleichswert
., c276 ea       nop
., c277 ea       nop
., c278 ea       nop
., c279 ea       nop
., c27a ea       nop
., c27b ea       nop
., c27c ea       nop
., c27d ea       nop
., c27e ea       nop
., c27f ea       nop
., c280 ea       nop
., c281 ea       nop
., c282 ea       nop
., c283 ea       nop
., c284 4c 93 c2 jmp $c293          weiter

., c287 bd 4d c3 lda $c34d,x        Ausgabe der Texte
., c28a f0 06    beq $c292
., c28c 20 d2 ff jsr $ffd2
., c28f e8       inx
., c290 d0 f5    bne $c287
., c292 60       rts

., c293 a9 0d    lda #$0d
., c295 20 d2 ff jsr $ffd2
., c298 a9 0d    lda #$0d
., c29a 20 d2 ff jsr $ffd2
., c29d a9 00    lda #$00
., c29f a2 c0    ldx #$c0
., c2a1 85 a7    sta $a7
., c2a3 86 a8    stx $a8
., c2a5 a9 00    lda #$00
., c2a7 a2 05    ldx #$05
., c2a9 85 a9    sta $a9
., c2ab 86 aa    stx $aa
., c2ad a9 08    lda #$08           LISTEN für Gerät Nummer 8
., c2af 20 b1 ff jsr $ffb1
., c2b2 a9 6f    lda #$6f           15; Kommandokanal
., c2b4 20 93 ff jsr $ff93
., c2b7 a9 4d    lda #$4d
., c2b9 20 a8 ff jsr $ffa8          Programm zur Floppy senden
., c2bc a9 2d    lda #$2d
., c2be 20 a8 ff jsr $ffa8
., c2c1 a9 57    lda #$57
., c2c3 20 a8 ff jsr $ffa8
., c2c6 a0 00    ldy #$00
., c2c8 a5 a9    lda $a9
., c2ca 20 a8 ff jsr $ffa8
., c2cd a5 aa    lda $aa
., c2cf 20 a8 ff jsr $ffa8
., c2d2 a9 1e    lda #$1e
., c2d4 20 a8 ff jsr $ffa8
., c2d7 b1 a7    lda ($a7),y
., c2d9 20 a8 ff jsr $ffa8
., c2dc c8       iny
., c2dd c0 1e    cpy #$1e
., c2df 90 f6    bcc $c2d7
., c2e1 20 ae ff jsr $ffae
., c2e4 18       clc
., c2e5 a5 a7    lda $a7
., c2e7 69 1e    adc #$1e
```

Listing 1. Eine neue Formatierroutine. Eine Diskette wird nicht nur schneller formatiert, sondern Sie können auch angeben, welche Spuren formatiert werden sollen. Was das für Vorteile hat, erfahren Sie im Bericht.

```
., c2e9 85 a7    sta $a7
., c2eb 90 03    bcc $c2f0
., c2ed e6 a8    inc $a8
., c2ef 18       clc
., c2f0 a5 a9    lda $a9
., c2f2 a6 aa    ldx $aa
., c2f4 69 1e    adc #$1e
., c2f6 85 a9    sta $a9
., c2f8 90 02    bcc $c2fc
., c2fa e6 aa    inc $aa
., c2fc e0 07    cpx #$07
., c2fe 90 ad    bcc $c2ad
., c300 c9 00    cmp #$00
., c302 90 a9    bcc $c2ad
., c304 a9 08    lda #$08           LISTEN für Gerät 8
., c306 20 b1 ff jsr $ffb1
., c309 a9 6f    lda #$6f           15; Kommandokanal
., c30b 20 93 ff jsr $ff93
., c30e a9 4d    lda #$4d
., c310 20 a8 ff jsr $ffa8
., c313 a9 2d    lda #$2d
., c315 20 a8 ff jsr $ffa8          Programm in der Floppy
., c318 a9 45    lda #$45           bei Adresse $0600 starten
., c31a 20 a8 ff jsr $ffa8
., c31d a9 60    lda #$60
., c31f 20 a8 ff jsr $ffa8
., c322 a9 06    lda #$06
., c324 20 a8 ff jsr $ffa8
., c327 20 ae ff jsr $ffae
., c32a a9 00    lda #$00
., c32c 85 90    sta $90
., c32e a9 08    lda #$08           Fehlermeldung holen
., c330 20 b4 ff jsr $ffb4
., c333 a9 6f    lda #$6f
., c335 20 96 ff jsr $ff96
., c338 20 a5 ff jsr $ffa5
., c33b 20 d2 ff jsr $ffd2          und anzeigen
., c33e 24 90    bit $90
., c340 50 f6    bvc $c338
., c342 20 ab ff jsr $ffab
., c345 4c dc c3 jmp $c3dc          Endebehandlung
.: C348 00 00 00 00 00 93 20 20
.: C350 20 20 20 20 20 2A 2A 2A
.: C358 20 44 49 53 4B 2D 46 4F
.: C360 52 4D 41 54 2D 53 59 53
.: C368 54 45 4D 20 2A 2A 2A 0D
.: C370 0D 0D 20 28 43 29 20 31
.: C378 39 38 35 20 42 59 20 4B
.: C380 4F 53 53 20 20 20 0D 0D
.: C388 0D 44 49 53 4B 4E 41 4D
.: C390 45 3A 20 00 0D 0D 44 49
.: C398 53 4B 2D 49 44 3A 20 00
.: C3A0 0D 0D 46 52 4F 4D 20 54
.: C3A8 52 41 43 4B 3A 24 00 0D
.: C3B0 0D 54 4F 20 54 52 41 43
.: C3B8 4B 3A 24 00 0D 0D 41 4E
.: C3C0 4F 54 48 45 52 20 46 4F
.: C3C8 52 4D 41 54 20 28 59 2F
.: C3D0 4E 29 20 3F 20 0D 0D 00
.: C3D8 00 00 00 00 20 29 C4 A2
., c3dc 20 29 c4 jsr $c429          SAVE-Vektor stellen
., c3df a2 6f    ldx #$6f
., c3e1 20 87 c2 jsr $c287          'ANOTHER FORMAT (Y/N) ?' ausgeben
., c3e4 20 e4 ff jsr $ffe4
., c3e7 f0 fb    beq $c3e4
., c3e9 c9 59    cmp #$59           Ende ?
., c3eb d0 03    bne $c3f0          ja: RTS
., c3ed 4c 00 c2 jmp $c200          noch einmal formatieren
., c3f0 60       rts
., c3f1 00       brk
., c3f2 a5 b7    lda $b7            Länge des Filenamens =0?
., c3f4 f0 03    beq $c3f9          ja; dann formatieren
., c3f6 4c ed f5 jmp $f5ed          zur SAVE-Routine
., c3f9 20 00 c2 jsr $c200          formatieren
., c3fc a9 01    lda #$01
., c3fe a2 00    ldx #$00
., c400 a0 00    ldy #$00
., c402 18       clc
., c403 60       rts                Ende

., c404 85 02    sta $02            Umrechnung der Eingabe in ein
., c406 86 03    stx $03            HEX-Byte
., c408 a5 02    lda $02
., c40a c9 41    cmp #$41
., c40c 90 03    bcc $c411
., c40e 18       clc
., c40f 69 09    adc #$09
., c411 29 0f    and #$0f
., c413 0a       asl
., c414 0a       asl
., c415 0a       asl
., c416 0a       asl
., c417 85 02    sta $02
., c419 a5 03    lda $03
., c41b c9 41    cmp #$41
., c41d 90 03    bcc $c422
., c41f 18       clc
., c420 69 09    adc #$09
., c422 29 0f    and #$0f
., c424 05 02    ora $02
., c426 85 02    sta $02
., c428 60       rts

., c429 a9 f2    lda #$f2           SAVE-Vektor herstellen
., c42b 8d 32 03 sta $0332
., c22e a9 c3    lda #$c3           auf Adresse $c3f2 setzen
., c230 8d 33 03 sta $0333
., c233 60       rts
```

Listing 1 (Schluß). Zwischen den Adressen C348 und C3DA liegt eine ASCII-Tabelle

Dieser Inhalt ist eigentlich auf einen Fehler im DOS zurückzuführen; er müßte, wie auch bei den großen Commodore-Floppies aus 256 $00-Bytes bestehen.

In meinem Programm fülle ich alle Sektoren mit dem üblichen Wert $00.

Noch ein paar Hinweise zur Benutzung des Formatierprogramms.

Nach RUN wird automatisch der SAVE-Vektor auf den Programmstart der Formatierroutine gestellt. Wird kein Filename angegeben, so erfolgt ein Sprung in das Formatierprogramm. Durch Drücken von RUN STOP/RESTORE läßt sich der SAVE-Vektor wieder richtig »hinbiegen«. Hierzu dürfte jedoch kein Anlaß bestehen, da bei fehlendem Filenamen kein Programm gestartet wird.

```
10 REM ************************        <137>
20 REM *                      *        <247>
30 REM *  DISK-FORMAT-SYSTEM  *        <052>
40 REM *                      *        <011>
50 REM *   (C) 1985 BY KOSS   *        <091>
60 REM *                      *        <031>
70 REM ************************        <197>
80 DATA 5657,5638,6947,7770,8264,7062,8578
   ,6111,3989,3215,9192,10797          <224>
90 DATA 8104,8232,8308,3524,3180,5204,4577 <144>
100 DATA 0,14,8,10,0,158,32,50,48,54,52,32
    ,32,0,0,0,162,64,160,8,134,2,132,3 <156>
110 DATA 162,0,160,192,134,4,132,5,160,0,1
    62,5,177,2,145,4,200,208,249,230,3 <187>
120 DATA 230,5,202,208,242,120,169,242,141
    ,50,3,169,195,141,51,3,88,96,234,234 <071>
130 DATA 165,10,201,36,144,7,169,18,133,67
    ,76,19,5,32,75,242,133,67,169,0,133 <043>
140 DATA 27,160,0,162,0,165,57,153,0,3,200
    ,200,165,27,153,0,3,200,165,10,153 <201>
150 DATA 0,3,200,165,19,153,0,3,200,165,18
    ,153,0,3,200,169,15,153,0,3,200,153 <254>
160 DATA 0,3,200,169,0,89,250,2,89,251,2,8
    9,252,2,89,253,2,153,249,2,230,27  <218>
170 DATA 165,27,197,67,144,190,169,3,133,4
    9,152,72,138,157,0,7,232,208,250,32 <092>
180 DATA 48,254,104,168,136,32,229,253,32,
    245,253,169,7,133,49,32,233,245,133 <100>
190 DATA 58,32,143,247,169,0,133,50,32,14,
    254,169,255,141,1,28,162,5,80,254  <251>
200 DATA 184,202,208,250,162,10,164,50,80,
    254,184,185,0,3,141,1,28,200,202,208 <125>
210 DATA 243,162,9,80,254,184,169,85,141,1
    ,28,202,208,245,169,255,162,5,80,254 <184>
220 DATA 184,141,1,28,202,208,247,162,187,
    80,254,184,189,0,1,141,1,28,232,208 <122>
230 DATA 244,160,0,80,254,184,177,48,141,1
    ,28,200,208,245,169,85,162,8,80,254 <146>
240 DATA 184,141,1,28,202,208,247,165,50,2
    4,105,10,133,50,198,27,208,149,80  <041>
250 DATA 254,184,80,254,184,32,0,254,169,2
    00,133,31,169,0,133,48,169,3,133,49 <160>
260 DATA 165,67,133,27,32,86,245,162,10,16
    0,0,80,254,184,173,1,28,209,48,208 <123>
270 DATA 14,200,202,208,242,24,165,48,105,
    10,133,48,76,53,6,198,31,208,209,169 <225>
280 DATA 6,76,211,253,32,86,245,160,187,80
    ,254,184,173,1,28,217,0,1,208,231  <087>
290 DATA 200,208,242,162,252,80,254,184,17
    3,1,28,217,0,7,208,215,200,202,208 <125>
300 DATA 241,198,27,208,176,76,158,253,160
    ,0,185,224,6,153,0,2,200,204,223,6 <150>
310 DATA 144,244,173,223,6,141,116,2,173,2
    22,6,141,123,2,169,0,133,127,32,0  <084>
320 DATA 193,172,123,2,185,0,2,133,18,185,
    1,2,133,19,32,7,211,169,26,141,5,28 <206>
330 DATA 169,192,133,0,165,0,48,252,174,22
    0,6,134,10,169,224,133,2,165,2,48  <128>
340 DATA 252,201,2,176,12,232,236,221,6,14
    4,236,32,64,238,96,234,234,162,2,76 <241>
350 DATA 10,230,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0    <204>
```

Listing 2. Der DATA-Lader der Formatierroutine



Mußten Sie dennoch einmal RESTORE drücken, so läßt sich das Formatier-System mit SYS 49664 ($C200) erneut starten; nach Beendigung wird unter anderem auch der SAVE-Vektor wieder auf das Programm zurückgestellt.

Wollen Sie sich den Disk-Status anzeigen lassen, so tippen Sie SYS49962. Es erscheint danach auch die Frage nach einem weiteren Formatiervorgang, die Sie entsprechend beantworten. Nach dieser Anzeige wird der SAVE-Vektor ebenfalls wieder hergestellt.

Ich möchte Sie an dieser Stelle auf ein paar Speicherstellen in der Zeropage der 1541 aufmerksam machen. Wie Sie wissen, werden dort nach einem RESET ein paar Konstanten abgelegt, die vom Benutzer (beliebig) verändert werden können. Mit den Konstanten meine ich zum Beispiel $08 als Kennzeichen eines Blockheaders oder $07 als Kennzeichen eines Datenblocks.

Wie Sie aus der Zeropage-Belegung in der 64'er-Ausgabe 1/1985, Seite 151 entnehmen können, werden diese beiden Werte in den Speicherstellen $38 (Wert 07) und $39 (Wert 08) abgelegt und können nun abgeändert werden. Der neue Wert, den Sie vielleicht in diese Speicherstellen eintragen, sollte sich jedoch im Bereich von $00 bis $0F bewegen, da es sonst Schwierigkeiten beim Lesen geben kann. Die Folge eines Leseversuchs mit normalen Werten, wenn eine Diskette anders formatiert wurde, sind entweder ein »20, READ ERROR« oder ein »22, READ ERROR«.

Der Vorteil dieser Errors ist jedoch die Möglichkeit, den Blöcken auch Inhalte mitzugeben, womit ein sehr wirkungsvoller Kopierschutz konstruiert werden kann.

Zum Lesen oder Beschreiben der Diskette müssen die Werte in den beiden Speicherstellen nur jeweils richtig gestellt werden; dann kann ein ganz normaler Zugriff stattfinden.

Mit Hilfe des Formatierprogramms können Sie jetzt auch noch zusätzlich illegale Spuren beschreiben. Hierbei müssen Sie allerdings, wie in der letzten Folge besprochen, auf Jobschleifenebene arbeiten, um die Begrenzung auf die Spuren 1 bis 35 zu umgehen.

In der nächsten Folge möchte ich Ihnen etwas vorstellen, das sich GCR-Codierung nennt. Wir werden dann erfahren, daß Daten keineswegs so auf Diskette geschrieben werden, wie sie im Puffer vorliegen, sondern, daß vorher eine Codierung erfolgt.

(Karsten Schramm/gk)

```
360 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0      <248>
370 DATA 162,0,32,135,194,160,0,32,207,255
    ,201,13,240,8,153,224,193,200,192   <149>
380 DATA 16,144,241,169,44,153,224,193,200
    ,140,222,193,162,71,32,135,194,162  <233>
390 DATA 0,32,207,255,201,13,240,9,153,224
    ,193,200,232,224,2,144,240,140,223  <206>
400 DATA 193,162,83,32,135,194,32,207,255,
    133,250,32,207,255,133,251,169,0,133 <097>
410 DATA 208,162,98,32,135,194,32,207,255,
    133,252,32,207,255,133,253,169,0,133 <114>
420 DATA 208,165,250,166,251,32,4,196,141,
    220,193,165,252,166,253,32,4,196,141 <125>
430 DATA 221,193,238,221,193,234,234,234,2
    34,234,234,234,234,234,234,234,234  <031>
440 DATA 234,234,76,147,194,189,77,195,240
    ,6,32,210,255,232,208,245,96,169,13 <123>
450 DATA 32,210,255,169,13,32,210,255,169,
    0,162,192,133,167,134,168,169,0,162 <100>
460 DATA 5,133,169,134,170,169,8,32,177,25
    5,169,111,32,147,255,169,77,32,168  <094>
470 DATA 255,169,45,32,168,255,169,87,32,1
    68,255,160,0,165,169,32,168,255,165 <165>
480 DATA 170,32,168,255,169,30,32,168,255,
    177,167,32,168,255,200,192,30,144   <050>
490 DATA 246,32,174,255,24,165,167,105,30,
    133,167,144,3,230,168,24,165,169,166 <204>
500 DATA 170,105,30,133,169,144,2,230,170,
    224,7,144,173,201,0,144,169,169,8   <039>
510 DATA 32,177,255,169,111,32,147,255,169
    ,77,32,168,255,169,45,32,168,255,169 <001>
520 DATA 69,32,168,255,169,96,32,168,255,1
    69,6,32,168,255,32,174,255,169,0,133 <009>
530 DATA 144,169,8,32,180,255,169,111,32,1
    50,255,32,165,255,32,210,255,36,144 <183>
540 DATA 80,246,32,171,255,76,220,195,0,0,
    0,0,0,147,32,32,32,32,32,32,32,42   <048>
550 DATA 42,42,32,68,73,83,75,45,70,79,82,
    77,65,84,45,83,89,83,84,69,77,32,42 <006>
560 DATA 42,42,13,13,13,32,40,67,41,32,49,
    57,56,53,32,66,89,32,75,79,83,83,32 <223>
570 DATA 32,32,13,13,13,68,73,83,75,78,65,
    77,69,58,32,0,13,13,68,73,83,75,45  <197>
580 DATA 73,68,58,32,0,13,13,70,82,79,77,3
    2,84,82,65,67,75,58,36,0,13,13,84   <155>
590 DATA 79,32,84,82,65,67,75,58,36,0,13,1
    3,65,78,79,84,72,69,82,32,70,79,82  <246>
600 DATA 77,65,84,32,40,89,47,78,41,32,63,
    32,13,13,0,0,0,0,0,32,41,196,162,111 <016>
610 DATA 32,135,194,32,228,255,240,251,201
    ,89,208,3,76,0,194,96,0,165,183,240 <011>
620 DATA 3,76,237,245,32,0,194,169,1,162,0
    ,160,0,24,96,133,2,134,3,165,2,201  <194>
630 DATA 65,144,3,24,105,9,41,15,10,10,10,
    10,133,2,165,3,201,65,144,3,24,105  <176>
640 DATA 9,41,15,5,2,133,2,96,169,242,141,
    50,3,169,195,141,51,3,96,0          <095>
1000 REM                                <122>
1010 REM **** DATAS INITIALISIEREN      <178>
1020 REM                                <142>
1030 RESTORE:PRINT:PRINT:PRINT"DATAS WERDE
     N UEBERPRUEFT !!!":PRINT:PRINT     <215>
1040 CLR:DIM P(19):DIM W(19)            <077>
1050 FOR X=0 TO 18:READ P(X):P=P+P(X):NEXT <139>
1060 IF P<>124349 THEN PRINT"PRUEFSUMMENFE
     HLER":PRINT:PRINT:LIST 80-90       <038>
1070 FOR X=0 TO 18:FOR Y=0 TO 59:READ A:W(
     X)=W(X)+A:NEXT Y                   <240>
1080 IF W(X)<>P(X)THEN 1150             <146>
1090 NEXT X                             <032>
1100 PRINT:PRINT"DIE DATAS SIND OK UND WER
     DEN":PRINT:PRINT"ABGESPEICHERT!"   <116>
1110 RESTORE:FOR X=0 TO 18:READ A:NEXT  <166>
1120 FOR X=0 TO 1139:READ A:POKE X+2048,A:
     NEXT                               <093>
1130 POKE 45,119:POKE 174,119:POKE 46,12:P
     OKE 175,12:CLR                     <006>
1140 PRINT:PRINT"MIT 'SAVE' ABSPEICHERN!":
     PRINT:END                          <097>
1150 REM FEHLERBEHANDLUNG               <159>
1160 PRINT:PRINT"FEHLER IN DEN DATAS"X*60"
      BIS"X*60+59" !":Z=INT(X*600/17.8) <161>
1170 PRINT:PRINT:PRINT"DAS ENTSPRICHT IN E
     TWA DEN{14SPACE,DOWN}ZEILEN AB"Z   <073>
1180 END                                <032>
```

Listing 2. Disk-Format-System (Schluß).

Fortsetzung von Seite 17

— Sonderfunktionen

Sie geben Auskunft über die Leistungsfähigkeit des Druckers. In Tabelle 2 sind einige Sonderfunktionen aufgeführt.

— Pufferspeicher

Verfügt ein Drucker über einen eingebauten Pufferspeicher, so kann das zweifachen Nutzen haben. Zum einen bieten manche Drucker eine Funktion an, mit der eigene Zeichensätze geladen werden können (Download-Charakters). Zum anderen verkürzt ein Pufferspeicher die Zeit, die Sie warten müssen, bis Sie den Computer weiterverwenden können, denn normalerweise reagiert kein Computer auf Eingaben, solange er Zeichen an den Drucker sendet.

— Geräuschpegel

Außer den Thermo- und Tintenstrahldruckern machen alle Drucker mehr oder weniger Lärm. Es ist deshalb ratsam sich ein neues Gerät nicht nur anzusehen, sondern auch anzuhören.

— Bedienungsanleitung

Sie ist von entscheidender Wichtigkeit für den Nutzen, den Sie aus Ihrem Drucker ziehen können. Eine Anleitung sollte umfassend und in deutscher Sprache über die Funktionen eines Drucker aufklären. An Beispielprogrammen sollte dabei die Funktionsweise des Druckers deutlich werden.

Jeder ist seines Druckers Schmied

Sie sehen, ein Druckerkauf ist Planungssache. Je nachdem auf welches Gebiet (Textverarbeitung, Programmierung oder Grafik) Sie Ihren Schwerpunkt setzen, wird Ihre Entscheidung anders ausfallen. Ein weiteres wichtiges Argument wird der Preis sein. Hier heißt es rechnen, denn mancher Drucker mit Interface kostet weniger als ein gleichwertiger Drucker ohne ein solches. Auch sollten Sie zukünftige Entwicklungen mit im Auge behalten, denn möglicherweise kaufen Sie einen neuen Computer, dann sollte am besten der alte Drucker weiterhin verwendbar sein. So ist ein Drucker mit Centronics-Schnittstelle sicher eine zukunftssichere Entscheidung, leider brauchen Sie für solch einen Drucker immer ein spezielles Interface. Ein Commodore-Drucker ist dagegen direkt anschließbar, paßt aber unter Umständen nicht an ihren nächsten Computer (es sei denn es ist der C 128). Obwohl gute Leistungen schon ab etwa 700 Mark zu haben sind, sollte bei einen Drucker ruhig etwas investiert werden, denn die Erfahrung zeigt, daß die Ansprüche schon nach kurzer Zeit steigen.

(Arnd Wängler/aa)


64ER ONLINE

Dem Klang auf der Spur (Teil 5)

Dieser Teil des Musikkurses ist auch für all jene interessant, die sich nicht ausschließlich für Musik interessieren. Es werden Algorithmen zur Generierung verschiedener Signale vorgestellt.

Dabei wird anhand des Source-Listings des Programms Modulator gezeigt, wie man diese Algorithmen unter zeitkritischen Nebenbedingungen programmieren kann.

Im zweiten Teil dieser Reihe wurde schon erwähnt, daß man jeden Signalverlauf durch eine Folge von Stützwerten beschreiben kann. Da man diese Stützwerte digital codieren kann, wird so die Signalerzeugung und -verarbeitung mit dem Computer möglich. Man muß dabei allerdings mit einer Abtastfrequenz arbeiten, die mindestens doppelt so hoch ist wie die höchste Frequenz, die im verarbeiteten Signal vorkommt. Für Audio-Signale in HiFi-Qulität ist somit eine Abtastfrequenz von mindestens 40 kHz erforderlich. Stellen wir dieser Frequenz einmal die Taktfrequenz von 1 MHz in unserem C 64 gegenüber: Eine Abtastperiode dauert bei 40 kHz 25 µs. Diese Zeit entspricht genau 25 Taktzyklen im C 64. Ein 6510-Maschinenbefehl dauert zwischen zwei und sieben Taktzyklen, das heißt, daß die CPU während einer Abtastperiode gerade vier bis maximal zwölf Befehle abarbeiten kann; zuwenig, um damit schon sinnvoll einen Signalabtastwert weiterzuverarbeiten. Die digitale Verarbeitung von Audiosignalen bleibt also zunächst einmal Hochleistungsrechnern und Spezialprozessoren vorenthalten.

Dreh- und Angelpunkt: Integer-Arithmetik

Wenn man sich aber wie bei dem in der letzten Folge vorgestellten Programm Modulator auf eine Abtastfrequenz von 60 Hz beschränkt, dann sieht die Sache schon sehr viel günstiger aus. Während einer Abtastperiode von 16,6 ms (entsprechend 16 600 Taktzyklen) kann man bei geeigneter Programmierung schon eine ganze Menge machen. Die Abtastfrequenz ist aber nicht das einzig wichtige Kriterium bei der Signalverarbeitung. Eine Rolle spielt auch die **Genauigkeit**, mit der die Abtastwerte dargestellt und verrechnet werden. Natürlich gilt hier: Je genauer, desto besser. Genauigkeit kostet aber wieder Rechenzeit, sobald man die Wortlänge des verfügbaren Prozessors (hier leider nur 8 Bit) überschreitet. Die Arithmetik-Routinen des Basic-Interpreters arbeitet zum Beispiel im Fließkommaformat mit 32-Bit-Mantisse. Sie bietet damit eine Genauigkeit, die selbst für sehr anspruchsvolle Probleme aus der Signalverarbeitung mehr als genug sein dürfte. Diese Routinen sind jedoch so langsam, daß sie auch in 16,6 ms nichts Vernünftiges tun können. Wir werden also unsere eigenen Arithmetik-Befehlsfolgen programmieren müssen und dabei einen Kompromiß zwischen Genauigkeit und Geschwindigkeit machen. Gleitkomma-Arithmetik ist zu aufwendig und für unsere Zwecke auch gar nicht erforderlich. Eine Genauigkeit von nur 8 Bit reicht allerdings auch nicht immer aus. So haben ja auch manche SID-Parameter eine Länge von 12 oder 16 Bit. Im Programm Modulator wird größtenteils mit 16-Bit-Zweierkomplex-Größen gerechnet. Es sei in diesem Zusammenhang auf dem Assembler-Kurs (Teil 3 im 64'er, Ausgabe 11/84) verwiesen, wo ausführlich beschrieben wird, wie man negative Zahlen im Zweierkomplement darstellt. Addition und Subtraktion von Zweierkomplement-Größen werden direkt durch die CPU-Befehle ADC und SBC sowie durch drei Flaggen (Negativ, Carry und Overflow) unterstützt. Für die Multiplikation gibt es dagegen keinen Maschinenbefehl. Da die Multiplikation in Modulator aber eine zentrale Rolle spielt, benötigen wir für sie ein effizientes (das heißt möglichst schnelles) Maschinenprogramm.

Die Multiplikation

Was ist 43 x 13? Die wenigsten Menschen dürften die Antwort auf einen Schlag parat haben, so wie zum Beispiel auf die Frage, was 3 x 7 ist. Wir brauchen 3 x 7 nicht auszurechnen, weil wir es auswendig wissen. Den Wert des Produkts 43 x 13 werden die wenigsten auswendig kennen und daher zu rechnen anfangen. Eine solche Rechnung könnte (ausführlich) so aussehen:

```
(1)     43 x 13
        -------
(2)     43
(3)         129
        -------
(4)         559
```

In Zeile (1) stehen dabei noch einmal die Faktoren. Zeile (2) stellt das Teilprodukt 43 x 10 = 430 dar, Zeile (3) das Teilprodukt 43 x 3 = 129. Durch geeignetes Einrücken braucht man die Null bei 430 in Zeile (2) nicht mitschreiben. In Zeile (4) werden schließlich die Teilprodukte addiert. Dieser vertrauten Rechenweise liegt das **Distributivgesetz** zugrunde, welches die Bildung von Teilprodukten erlaubt:

(5) 43x(10+3)=(43x10)+(43x3)

Die »komplizierte« Multiplikation 43 x 13 wird also auf »einfachere« Multiplikationen 43 x 10 und 43 x 3 zurückgeführt. Bezeichnen wir in unserem Beispiel die Zahl 43 als **Multiplikant (MD)** und 13 als **Multiplikator (MR)**, so können wir das Multiplikationsschema so formulieren:

»Multipliziere MD mit den einzelnen Dezimalstellen von MR und addiere die Teilproduktion, die mit den entsprechenden Zehnerpotenzen 1,10,1000 etc. zu skalieren sind. Die Skalierung erreicht man aber einfach durch Linksverschieben um 0,1,2 etc. Dezimalstellen.«

Man kann den Multiplikator aber auch anders zerlegen, zum Beispiel in Zweierpotenzen:

13 = 8 + 4 + 1

und so multiplizieren:

```
(6)  43x13 = 8x43 + 4x43 + 1x43
           = 344  + 172  +  43
           = 599
```

Man benötigt hier als Summanden Produkte von MD mit Zweierpotenzen, die man leicht durch wiederholtes Verdoppeln von MD erhalten kann. Auf diese Weise sollen übrigens schon die alten Ägypter multipliziert haben. Wenn man nun MD und MR im Binärsystem darstellt, kann man besonders einfach multiplizieren: Die Produkte von MD mit Zweierpotenzen erhält man ganz einfach durch wiederholtes Linksverschieben. In unserem Beispiel gilt in binärer Schreibweise: MD = 101011, MR = 1101. Es ergibt sich das Schema:

```
             MD       MR
(7)        101011 x 1101
           -------------
(8)        101011
(9)         101011
(10)   +      101011
           -------------
(11)       1000101111
```

In Zeile (7) stehen MD und MR. Die Zeilen (8), (9) und (10) entsprechen den Teilpodukten 43 x 8, 43 x 4 und 43 x 1, die durch Linksverschiebung aus MD hervorgehen. Die Summe in Zeile (11) ist genau die Binärdarstellung von 559 (nachrechnen!).

Nach diesem Schema kann man nun einen Algorithmus formulieren: Es sei dazu N die Zahl der Binärstellen von MR:

```
1 SUM:=0;
2 FOR I:=N–1 DOWNTO 0 DO
3 BEGIN
4 SUM := LINKS(SUM);
5 IF MR(I)=1 THEN SUM:=SUM+MD
6 END
```

Der Algorithmus ist hier formal in einem Pascal-ähnlichen Stil dargestellt. SUM wird zunächst mit 0 vorbesetzt und dann N-mal nach links geschoben. Immer wenn dabei, von links nach rechts gezählt, in MR eine Eins auftritt, wird MD zu SUM addiert.

Sehen wir uns für unser Beispiel einen Trace des Programms an:

MD = 101011 MR = 1101 N = 4

Zeile	I	MR(I)=1	SUM	AKTION
1	undef	undef	0	
4	3	true	0	LINKS
5	3	true	101011	+
4	1	true	1010110	LINKS
5	2	true	10000001	+
4	1	false	100000010	LINKS
5	1	false	100000010	nichts
4	0	true	1000000100	LINKS
5	0	true	1000101111	+

Wir wollen diesen Algorithmus nun konkret in Maschinensprache realisieren. Wenn MD und MR zunächst auf 8 Bit begrenzt werden, kann man auf sie mit einem einzigen Maschinenbefehl zugreifen. Die Variable SUM hält man am besten im Akkumulator, weil man sie zum Addieren sowieso dorthin laden müßte. Auch die Linksverschiebung des Akkumulators ist wesentlich schneller als die einer Speicherzelle. Beim Addieren in den Akkumulator und beim Linksverschieben treten allerdings Überträge auf, die nicht verloren gehen dürfen, da diese je gerade die höherwertigen Bits von SUM darstellen. Das Endprodukt, das sich in SUM bildet, kann bis zu 16 Bit lang werden. (In unserem Beispiel sind es immerhin schon 9 Bit.) Bild 1 zeigt eine elegante Realisierung des Algorithmus, die mit nur zwei Speicherplätzen auskommt.

Die langen Rechtecke stellen die Speicherstellen MD, MR und

den Akkumulator dar, fette Linien stehen für Bytepfade, dünne für Bitpfade. »+« versinnbildlicht die Addition A := A + MD. In diesem Schema werden MR und A zusammen (wie ein 16-Bit-Register) nach links geschoben. Dadurch erscheinen die Bits von MR nacheinander in der Carry-Flagge und können so leicht abgefragt werden. MD wird nur dann zum Akku addiert, wenn das durch Linksverschiebungen aus MR gewonnene Bit Eins ist. Ein Übertrag bei der Addition in den Akku muß natürlich nach MR weitergegeben werden, da MR gleichzeitig auch die höherwertigen Bits von SUM enthält. Durch die doppelte Nutzung der Speicherstelle MR wird der Multiplikator zwar durch das höherwertige Byte von SUM überschrieben, man spart sich dadurch aber einen Schiebebefehl und einen Speicherplatz. Da wir wegen schnelleren Zugriffs MD und MR in der Zero-Page plazieren werden, und da der freie Platz dort knapp ist, ist die Einsparung von Speicherplatz durchaus gerechtfertigt. Hier das Programm, das ausführlich besprochen werden soll:

```
MUL  LDA #0       (2)   SUM:=0
     LDX #8       (2)   Schleifen-
                        zähler
LOOP ASL A        (2)   Register-
                        paar
     ROL MR       (5)   (MR.A)
                        n. links
     BCC NEXT     (2/3)
     CLC          (2)
     ADC MD       (3)   SUM:=
                        SUM+MD
     BCC NEXT     (2/3)
     INC MR       (5)   Übertrag
                        nach MR
NEXT DEX          (2)
     BNE LOOP     (2/3) nächster
                        Lauf
```

Die Zahlen in Klammern geben die Ausführungszeiten der Befehle in Taktzyklen an. Sie sind aus Tabelle 1 entnommen. Bei den Verzweigungen nehmen wir der Einfachheit halber an, daß keine Page-Grenzen übersprungen werden, sonst müßte man im Falle eines Sprunges vier statt drei Takte in Rechnung stellen. Zur Arbeitsweise des Programms: Zuerst wird SUM mit 0 vorbesetzt. Dazu genügt es, den Akku mit 0 zu besetzen, da die höherwertigen Bits von SUM erst durch den Schiebeprozeß entstehen. Das X-Register zählt die Schleifendurchläufe. Innerhalb der Schleife wird zunächst das Registerpaar (MR.A) durch das Befehlspaar ASL,ROL nach links verschoben. Beide Befehle schieben nach links, wobei Bit 7 in die Carry-Flagge geschoben wird. Der Unterschied der beiden Befehle besteht aber darin, daß ASL das Bit 0 immer mit Null besetzt, während ROL Bit 0 mit dem Wert besetzt, den die Carry-Flagge vor dem ROL-Befehl hatte. In unserem Fall ist das gerade das aus dem vorhergehenden ASL stammende Bit 7 vom Akku. Nach der Verschiebung zeigt das aus MR stammende Carry-Bit an, ob MD zu SUM addiert werden soll oder nicht. Falls nicht, wird mit BCC NEXT die Addition übersprungen. Die Addition selbst berücksichtigt durch ein weiteres BCC NEXT/INC MR einen eventuell auftretenden Übertrag nach MR. Die Speicherstelle MR enthält zwar gleichzeitig Teile vom Multiplikator und von SUM, man kann sich aber überlegen, daß ein Übertrag nach MR nur den SUM-Teil, aber nicht den Multiplikator-Teil beeinflußt. Nach dem Verlassen der Schleife steht schließlich das 16-Bit-Produkt (die Variable SUM) im Registerpaar (MR.A). Der Multiplikator wurde überschrieben, der Multiplikant in MD dagegen ist unverändert erhalten geblieben.

			Immediate	Zero Page	ZP,X ZP,Y Abs.	Abs., X/Y	(Ind, X)	(Ind), Y
LDA LDX LDY ADC	AND ORA EOR SBC	BIT CMP CPX CPY	2	3	4	4/5*	6	5/6*
	STA STX STX		—	3	4	5	6	6

			Akku	Zero Page	ZP,X Abs.	Abs.,X
ASL LSR	ROL ROR	INC DEC	2	5	6	7

*die größere Zahl gilt, wenn beim Indizieren eine Page-Grenze überschritten wird

	Absolut	(Indirekt)
JMP	3	5
JSR	6	—

				Relativ
BCC BCS	BEQ BNE	BMI BPL	BVC BVS	2/3/4**

						Implizit
CLC CLD CLI	SEC SED SEI	DEX DEY INX INY	TAX TAY TSX	TXA TYA TXS	NOP	2
		PHA	PHP			3
		PLA	PLP			4
		RTI	RTS			6

**2, wenn nicht gesprungen wird
3, wenn gesprungen wird und das Sprungziel auf der gleichen Page liegt
4, wenn auf ein Ziel in einer anderen Page gesprungen wird

Tabelle 1. Ausführungszeiten der 6502/6510-Maschinenbefehle (in Taktzyklen)

Zeitbedarf

Es soll hier exemplarisch gezeigt werden, wie man den genauen Zeitbedarf eines Maschinenprogramms ermittelt. Die Ausführungszeit des Multiplikationsprogramms ist nicht einheitlich. Sie hängt von den Anfangswerten von MR und MD ab, welche das Verhalten des Programms an den beiden Verzweigungsstellen (BCC NEXT) beeinflussen. Wir werden hier also den günstigsten (in bezug auf die Rechenzeit) und den ungünstigsten Fall untersuchen. Im ungünstigsten Fall muß bei jedem Schleifendurchlauf addiert werden, und zusätzlich tritt bei jeder Additon ein Übertrag auf. Der Zeitbedarf eines Schleifendurchlaufes beträgt dann:

2(ASL) + 5(ROL) + 2(BCC) + 2(CLC) + 3(ADC) + 2(BCC) + 5 (INC) + 2(DEX) + 3(BNE) = = 26 Takte

Die Gesamtdauer der Multiplikation ergibt sich dann so: 2(LDA) + 2(LDX) + 8*26(Schleife) -1 = **211**

Die -1 kommt dadurch zustande, daß beim letzten Schleifendurchlauf bei BNE nicht gesprungen wird und dadurch nur 2 statt 3 Takte benötigt werden.

Im günstigsten Fall (MR = 0, die Addition wird immer übersprungen) braucht die Schleife:

2(ASL) + 5(ROL) + 3(BCC) + 2(DEX) + 3(BNE) = = 16 Takte

Gesamtdauer:

2(LDA) + 2(LDX) + 8*16 (Schleife)-1 = **131**

Die Ausführungszeit der Multiplikation liegt also immer zwischen 131 und 211 Takten, wobei die Grenzwerte wohl selten erreicht werden dürften. Man kann im Mittel wohl mit zirka 170 Takten rechnen. Es ist für unsere Zwecke sehr wichtig, diese Größe zu kennen. Wenn wir in unserem Programm Modulator für einen Schritt eine Zeit von maximal 16,6 ms zur Verfügung haben, so können wir daraus eine theoretische Obergrenze für die Anzahl der in einem Schritt ausführbaren Multiplikationen ableiten. Sie liegt bei unserer 8-mal-8-Bit-Multiplikation etwa bei 75, wenn man den ungünstigsten Fall zugrundelegt.

Multiplikation mit größerer Wortlänge

Bei Modulator wird die Multiplikation für folgende Zwecke benötigt: Die LFOs und der Hüllkurvengenerator erzeugen Werteverläufe mit maximaler Amplitude. Das bedeutet bei der 16-Bit-Zweierkomplement-Arithmetik, in der hauptsächlich gerechnet wird, daß die Werte den zur Verfügung stehenden Bereich von -32768 bis +32767 meistens voll ausschöpfen. Nun möchte man aber oft das Modulationsziel, zum Beispiel die Frequenz einer SID-Stimme, nur um einige Hertz nach oben und unten modulieren. Man möchte die Tiefe dieser Modulation aber auch möglichst kontinuierlich steuern können, so daß zum Beispiel auch Modulationstiefen von einer Quinte oder gar einer Oktave möglich sind. Aus diesem Grund muß das Modulationssignal erst mit einem geeigneten Skalierungsfaktor multipliziert werden. Anschließend kann es durch einfache Addition zur Zielgröße diese in dem gewünschten Sinn modulieren.

Bei der Modulation von Tonhöhen ergibt sich außerdem noch

ein weiteres Problem: Dort kommt es nicht auf absolute, sondern auf relative Frequenzverschiebungen an. Ein Beispiel: Ein 500-Hz-Ton wird um ±5 Hz moduliert. Um bei einem 1000-Hz-Ton den gleichen Effekt zu erzielen, muß man ihn um ±10 Hz modulieren. Der Modulationsbetrag muß also bei Tonhöhen zusätzlich mit der zu modulierenden Frequenz selbst skaliert werden, was eine weitere Multiplikation erforderlich macht.

Die Wortlängen der Modulationsziele sind:

Tonfrequenzen	16 Bit
Pulsweiten	12 Bit
Filterfrequenz	8 Bit
Lautstärke	4 Bit

Die Filterfrequenz ist beim SID zwar eine 11-Bit-Größe, da aber feine Frequenzunterschiede in der Filterfrequenz nicht hörbar sind, werden nur die oberen 8 Bit moduliert.

Zur Steuerung der Modulationstiefe genügen 8-Bit. Eine fein gestufte Modulation ist ohnehin nur bei der Tonhöhenmodulation erforderlich. Hier genügt es aber, wenn das Modulationssignal selbst einen fein gestuften Verlauf (16 Bit) hat. Die durch 8 Bit realisierbaren 255 verschiedenen Modulationstiefen reichen aus, um alles vom feinsten Vibrato über Tonhöhensprünge in allen musikalisch sinnvollen Intervallen bis hin zur Sirene mit weitem Frequenzbereich zu verwirklichen.

Wir benötigen also eine 16 x 8-Bit-Multiplikation. Der Algorithmus von Bild 1 ließe sich in diese Richtung leicht erweitern. Man kann etnweder MR auf 16 Bit verlängern und benötigt dann 16 statt 8 Schleifendurchläufe oder man verlängert MD und A auf 16 Bit. In letzterem Fall benötigt man weiterhin nur 8 Schleifendurchläufe wobei aber, im Falle einer Eins aus MR, zwei 16-Bit-Größen (MD und A) addiert werden müssen. Natürlich braucht man für das höherwertige Byte von A einen weiteren Speicherplatz in der Zero-Page.

In Modulator wird ein anderer Weg eingeschlagen. Er wird durch Bild 2 beschrieben. Dieses erscheint zwar zunächst sehr kompliziert, das zugehörige Programm benötigt aber eine geringere Ausführungszeit. Vorgegeben sind ein 16-Bit-Multiplikator im Registerpaar (MR + 1.MR) und ein 8-Bit-Multiplikant in MD. (Der Einfachheit halber werden hier Zero-Page-Speicherplätze »Register« genannt.) Das Ergebnis des Programms soll ein 24-Bit-Produkt im Register Tripel (MR + 1.MR.A) sein.

Zuerst wird das niederwertige Teilprodukt MR x MD gebildet. Das Rechteck mit dem Kreuz steht für das Verfahren aus Bild

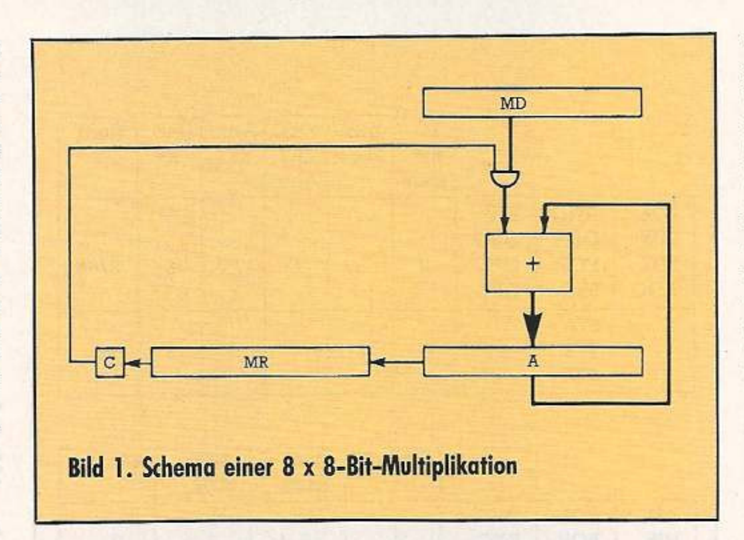


Bild 1. Schema einer 8 x 8-Bit-Multiplikation

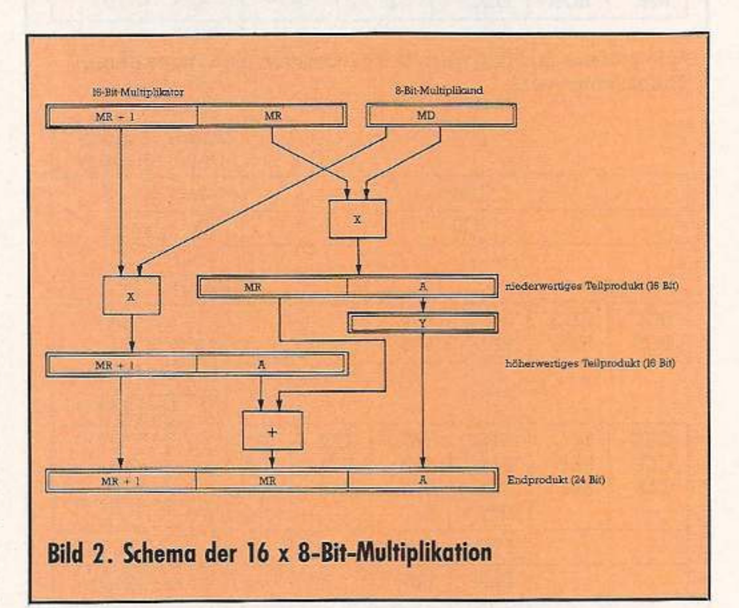


Bild 2. Schema der 16 x 8-Bit-Multiplikation

1, welche wie schon beschrieben, ein 8 x 8-Bit-Produkt in (MR.A) liefert. A wird im Y-Register zwischengspeichert. Anschließend werden MR+1 und MD ebenfalls nach Bild 1 multipliziert. Das Ergebnis ist das höherwertige Teilprodukt in (MR+1.A). Schließlich müssen die Teilprodukte nur noch mit richtiger Skalierung addiert werden. Dazu wird das höherwertige Byte des niederwertigen Teilprodukts, das in MR steht, zum niederwertigen Byte des höherwertigen Teilprodukts, das sich schon im Akku befindet, addiert. Dabei muß ein eventueller Übertrag nach MR+1 berücksichtigt werden. Das niederwertige Byte des niederwertigen Teilprodukts wird nur noch vom Y-Register in den Akku übertragen, wo es den niederwertigsten Teil des Endprodukts darstellt. In Modulator werden allerdings grundsätzlich nur 16-Bit-Größen weiterverarbeitet, so daß diese untersten 8 Bit des Produktes unberücksichtigt bleiben.

Im Source Listing zu Modulator steht das zugehörige Programm MULU in den Zeilen 1680 bis 1970. Zunächst steht dort zweimal hintereinander das schon vorgestellte 8 x 8 Bit-Multiplikationsprogramm, anschließend werden ab Zeile 1910 die Teilprodukte addiert. Eine Analyse ergibt eine Laufzeit von minimal **282** Takten und maximal **446** Takten.

Alle bisher beschriebenen Multiplizierer arbeiten nur dann korrekt, wenn man die Faktoren als positive Ganzzahlen interpretiert. Sie sind ohne Ergänzung nicht für Zweierkomplement-Größen geeignet. Das Programm MULS ab Zeile 2020 ist eine solche Ergänzung. Es berücksichtigt das Vorzeichen des Multiplikators. Ist dieser positiv, so wird sofort nach MULU verzweigt. Ein negativer Multiplikator wird zunächst negiert, wodurch er positiv wird (Zeile 2040 bis 2100), MULU wird als Unterprogramm aufgerufen, und schließlich wird das positive Produkt noch einmal negiert, was dann ein korrektes Resultat liefert. Der 8-Bit-Multiplikant wird aber nach wie vor nur als positive Zahl behandelt.

Die LFOs

Sie erzeugen die für Modulationen sinnvollen Kurvenverläufe als Folge von 16-Bit-Zweierkomplement-Zahlen. Am häufigsten wird die Dreieckskurve benötigt, da sie keine Sprünge macht und daher bei Anwendung auf Tonhöhen und auf Pulsweiten am angenehmsten klingt. Der Sägezahn eignet sich mehr für »härtere Effekte und für Videospiele, wo stark und schnell modulierte Töne oft zu hören sind. Die Rechteckkurve eignet sich für Triller (bei Frequenzmodulation), für mandolinenartige Effekte (bei Modulation von Lautstärke und Filterfrequenz) sowie für rhythmische Effekte (bei Frequenzmodulation mit größerer Modulationstiefe).

Rechnerisch kann man einen Sägezahnförmigen Wertverlauf besonders einfach erzeugen. Bei Modulator wird einfach ein 16-Bit-Wert zyklisch hochgezählt. Zyklisch bedeutet, daß immer wieder beim Minimalwert angefangen wird, wenn der Maximalwert überschritten wird. Das geschieht bei begrenzter Wortlänge automatisch durch Überlauf, den man hier absichtlich unberücksichtigt läßt. Im Modulator-Programm wird der Werteverlauf durch das Wort (= Bytepaar) SAWUP repräsentiert. SAWUP wird einfach um den Betrag im Wort LFOF hochgezählt. Dadurch ist die resultierende Frequenz der Sägezahnkurve direkt proportional zum Wert LFOF. Im Programm wird in Zeile 2300 bis 2420 erst das Steuerregister LFOC abgefragt. Im Falle des HOLD- oder RESET-Status braucht nichts berechnet zu werden. Im Falle des RUN-Status wird SAWUP in den Zeilen 2430 bis 2510 hochgezählt. Der aufsteigende Sägezahn wird dann gewissermaßen als »Master« für die anderen Kurvenformen herangezogen. Bild 3 zeigt, wie diese aus SAWUP gewonnen werden.

Interessant ist, daß gleichgültig, ob man die SAWUP-Werte im Zweierkomplement oder grundsätzlich positiv interpretiert, sich immer der gleiche Kurvenverlauf ergibt (gestrichelte und durchgezogene Kurve bei SAWUP).

Die Rechteckkurve entsteht dadurch, daß man, gesteuert durch SAWU, zwischen den Extremwerten +LFOA x 2 (hoch) 7 und -LFOA x 2 (hoch) 7 hin- und herschaltet. Man spart sich so die sonst anschließend fällige Multiplikation mit der LFO-Amplitude (= Modulationstiefe) LFOA. Hin- und hergeschaltet wird, wenn der Sägezahnwert einen vorgegebenen Schwellwert über- beziehungsweise unterschreitet. Dieser Schwellwert ist nichts anderes als die Pulsweite LFOP. SAWDOWN erhält man einfach durch Negieren von SAWUP. Bildet man das Maximum von SAWUP und SAWDOWN, so erhält man einen dreieckförmi-

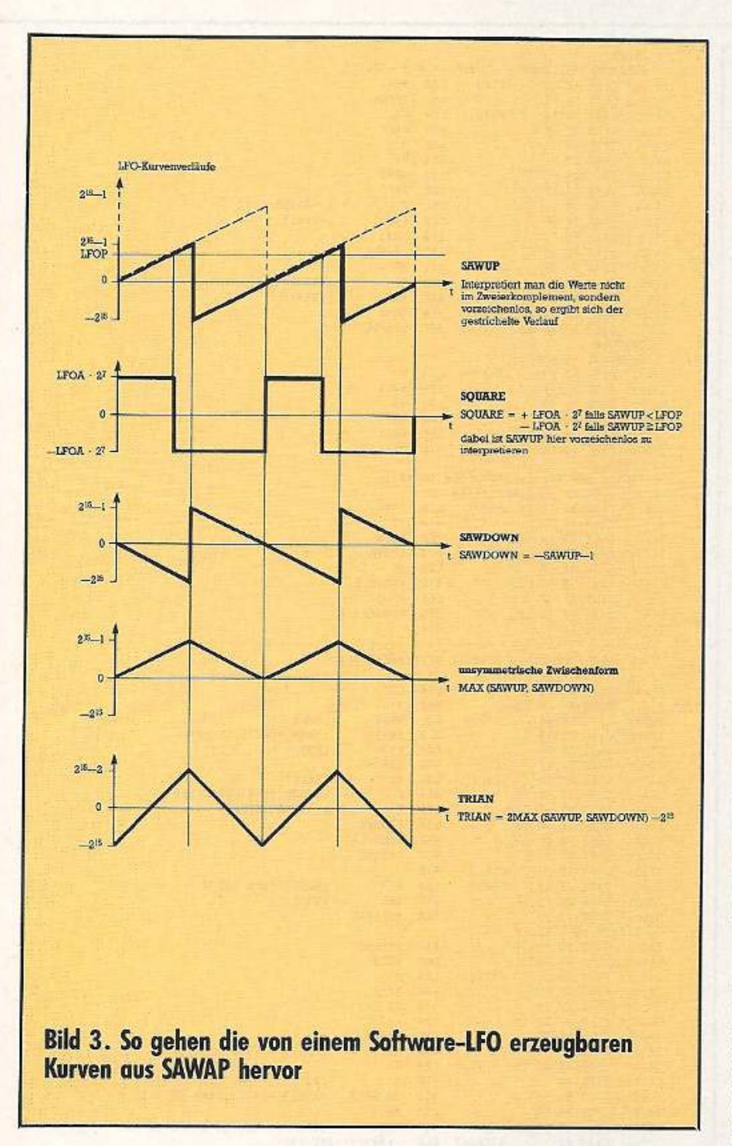
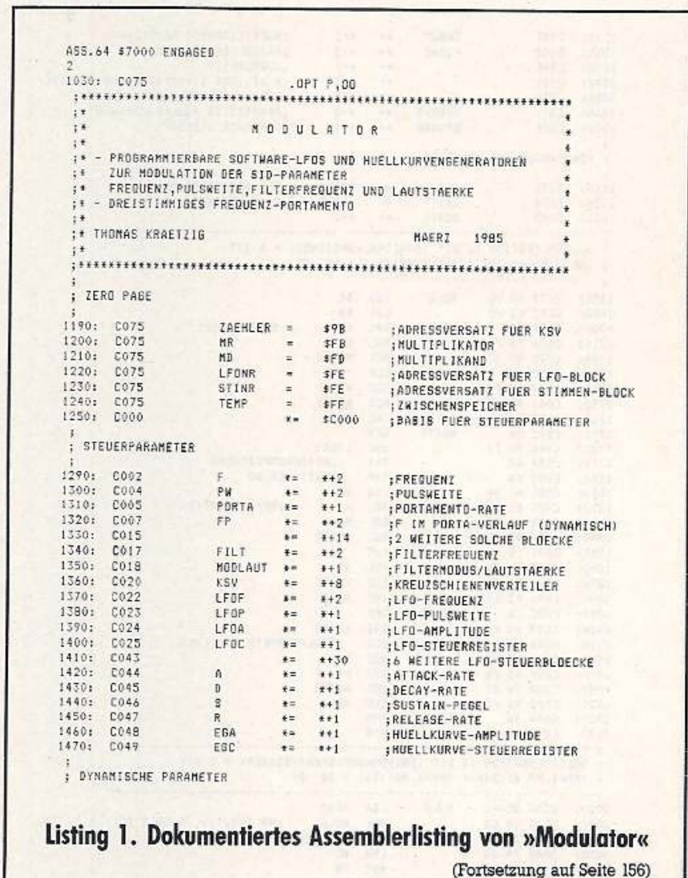


Bild 3. So gehen die von einem Software-LFO erzeugbaren Kurven aus SAWAP hervor



```
ASS.64 $7000 ENGAGED
2
1030:  C075                    .OPT P,00
 ;*****************************************************************
 ;*                                                               *
 ;*                    M O D U L A T O R                          *
 ;*                                                               *
 ;* - PROGRAMMIERBARE SOFTWARE-LFOS UND HUELLKURVENGENERATOREN    *
 ;*   ZUR MODULATION DER SID-PARAMETER                            *
 ;*   FREQUENZ,PULSWEITE,FILTERFREQUENZ UND LAUTSTAERKE           *
 ;* - DREISTIMMIGES FREQUENZ-PORTAMENTO                            *
 ;*                                                               *
 ;* THOMAS KRAETZIG                          MAERZ   1985         *
 ;*                                                               *
 ;*****************************************************************
 ;
 ; ZERO PAGE
 ;
1190:  C075          ZAEHLER  =   $9B     ;ADRESSVERSATZ FUER KSV
1200:  C075          MR       =   $FB     ;MULTIPLIKATOR
1210:  C075          MD       =   $FD     ;MULTIPLIKAND
1220:  C075          LFONR    =   $FE     ;ADRESSVERSATZ FUER LFO-BLOCK
1230:  C075          STINR    =   $FE     ;ADRESSVERSATZ FUER STIMMEN-BLOCK
1240:  C075          TEMP     =   $FF     ;ZWISCHENSPEICHER
1250:  C000                   *=  $C000   ;BASIS FUER STEUERPARAMETER
 ;
 ; STEUERPARAMETER
 ;
1290:  C002          F        *=  *+2     ;FREQUENZ
1300:  C004          PW       *=  *+2     ;PULSWEITE
1310:  C005          PORTA    *=  *+1     ;PORTAMENTO-RATE
1320:  C007          FP       *=  *+2     ;F IM PORTA-VERLAUF (DYNAMISCH)
1330:  C015                   *=  *+14    ;2 WEITERE SOLCHE BLOECKE
1340:  C017          FILT     *=  *+2     ;FILTERFREQUENZ
1350:  C018          MODLAUT  *=  *+1     ;FILTERMODUS/LAUTSTAERKE
1360:  C020          KSV      *=  *+8     ;KREUZSCHIENENVERTEILER
1370:  C022          LFOF     *=  *+2     ;LFO-FREQUENZ
1380:  C023          LFOP     *=  *+1     ;LFO-PULSWEITE
1390:  C024          LFOA     *=  *+1     ;LFO-AMPLITUDE
1400:  C025          LFOC     *=  *+1     ;LFO-STEUERREGISTER
1410:  C043                   *=  *+30    ;6 WEITERE LFO-STEUERBLOECKE
1420:  C044          A        *=  *+1     ;ATTACK-RATE
1430:  C045          D        *=  *+1     ;DECAY-RATE
1440:  C046          S        *=  *+1     ;SUSTAIN-PEGEL
1450:  C047          R        *=  *+1     ;RELEASE-RATE
1460:  C048          EGA      *=  *+1     ;HUELLKURVE-AMPLITUDE
1470:  C049          EGC      *=  *+1     ;HUELLKURVE-STEUERREGISTER
 ;
 ; DYNAMISCHE PARAMETER
```

Listing 1. Dokumentiertes Assemblerlisting von »Modulator«

(Fortsetzung auf Seite 156)

gen Kurvenverlauf, der allerdings nur positive Werte annimmt. Durch Verdoppeln dieser Werte und Verschiebung um 2(hoch) 15 nach unten erhält man dann eine symmetrische Dreieckskurve maximaler Amplitude.

Im Programm wird in Zeile 2530 bis 2590 aus LFOC ermittelt, welche Kurvenform überhaupt erzeugt werden soll, und entsprechend weiterverzweigt. Mit Ausnahme des Rechtecks, das schon mit seiner endgültigen Amplitude aufwartet, wird der errechnete Wert noch mit der Amplitude LFOA multipliziert (Zeile 3090 bis 3170). Das LFO-Programm rechnet alle 7 LFOs. Dabei wird auf die jeweiligen Parameter indiziert zugegriffen. Das Byte LFONR enthält dazu einen Adreß-Offset, der vom LFO-Programm in das X-Register geladen wird. Dieser Offset muß vom Programm, welches das LFO-Programm aufruft, korrekt zur Verfügung gestellt werden.

Aliasing-Parasitäre Frequenzen

Bei der eben beschriebenen Erzeugung der LFO-Kurvenformen tritt bei etwas höheren Frequenzen, etwa ab 10 Hz, noch ein interessantes Phänomen auf. Man kann die Erzeugung eines Sägezahnverlaufs durch zyklisches Hochzählen eines Wortes auch als Abtastung einer hypothetischen, kontinuierlichen Sägezahnkurfe auffassen. Die Abtastfrequenz ist in unserem Fall mit 60 Hz fest. Die Frequenz der hypothetischen Sägezahnkurve kann man aber durch den Parameter LFOF sehr feinstufig zwischen 0 und 60 Hz variieren. Wie soll aber zum Beispiel eine LFO-Kurve mit 50 Hz aussehen, wenn

Fortsetzung auf Seite 173

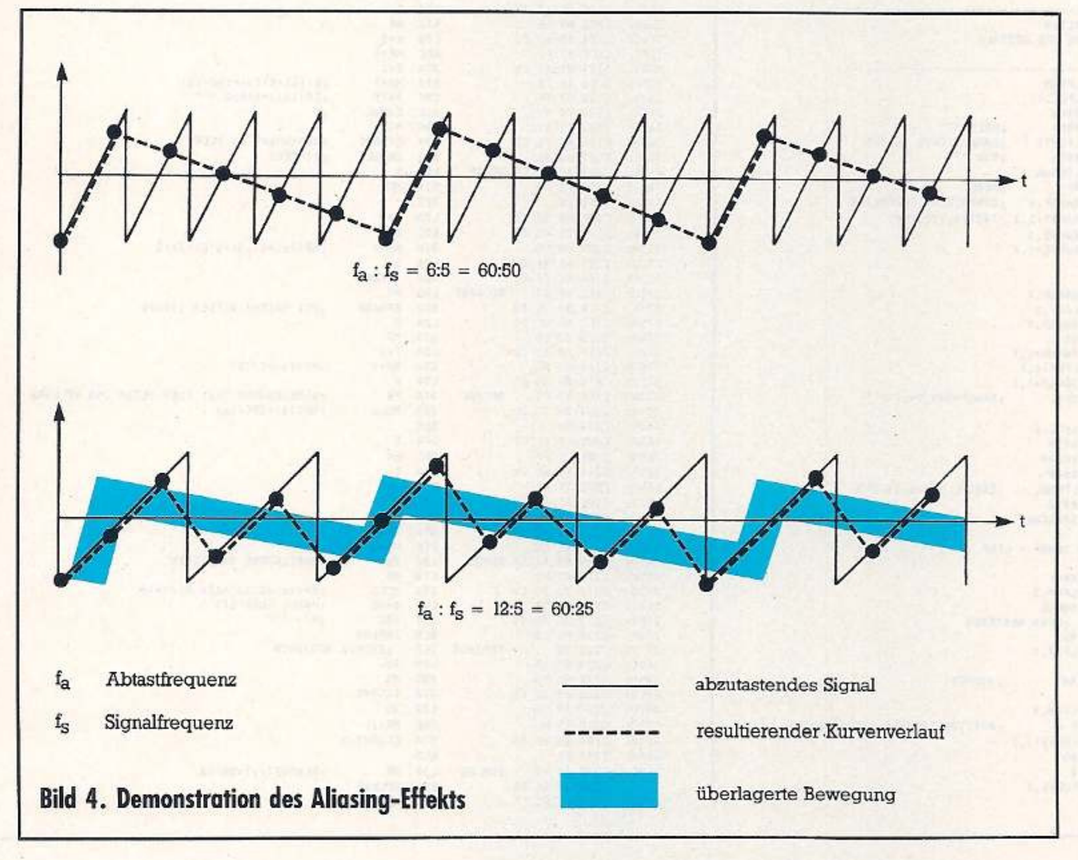


Bild 4. Demonstration des Aliasing-Effekts

```
;
C04B         SAWUP    *=   *+2      ;AUFSTEIGENDER SAEGEZAHN
C04D         KURVE    *=   *+2      ;AKTUELLER LFO-WERT
C04E                  *=   *+1      ;(UNBENUTZT)
C06C                  *=   *+30     ;6 WEITERE DYNAMISCHE LFO-BLOECKE
C06E         E        *=   *+2      ;HUELLKURVENWERT
C070         EKURVE   *=   *+2      ;BEWERTETER HUELLKURVENWERT
C071         EPHASE   *=   *+1      ;0=ATTACK 1=DECAY
;
; KONSTANTEN UND SONSTIGE
;
C071         SID      =    $D400    ;SID-BASISADRESSE
C073         ZEIT     *=   *+2      ;FUER ZEITMESSUNG
C075         ZEIT1    *=   *+2
;----------------------------------------------------------------
; MULTIPLIKATION 16 BIT (GANZZAHL=UNSIGNED) * 8 BIT
; (MR+1.MR.A)(24) = (MR+1.MR)(16) * MD (8)
;------------------------------
C075 A9 00   MULU     LDA  #0
C077 A2 08            LDX  #8
C079 0A      LOOP1    ASL  A        ;MR(8)*MD(8)
C07A 26 FB            ROL  MR
C07C 90 07            BCC  NEXT1
C07E 18               CLC
C07F 65 FD            ADC  MD
C081 90 02            BCC  NEXT1
C083 E6 FB            INC  MR
C085 CA      NEXT1    DEX
C086 D0 F1            BNE  LOOP1
C088 A8               TAY   ;ZWISCHENSPEICHERN
C089 8A               TXA   ;STATT LDA #0
C08A A2 08            LDX  #8
C08C 0A      LOOP2    ASL  A        ;MR+1(8)*MD(8)
C08D 26 FC            ROL  MR+1
C08F 90 07            BCC  NEXT2
C091 18               CLC
C092 65 FD            ADC  MD
C094 90 02            BCC  NEXT2
C096 E6 FC            INC  MR+1
C098 CA      NEXT2    DEX
C099 D0 F1            BNE  LOOP2
C09B 18               CLC   ;TEILPRODUKTE ADDIEREN
C09C 65 FB            ADC  MR
C09E 85 FB            STA  MR
C0A0 90 02            BCC  NEXT3
C0A2 E6 FC            INC  MR+1
C0A4 98      NEXT3    TYA
C0A5 60               RTS
;----------------------------------------------------------------
; MULTIPLIKATION 16 BIT (ZWEIERKOMPLEMENT=SIGNED) * 8 BIT
; (MR+1.MR.A)(24) = (MR+1.MR)(16) * MD (8)
;----------------------------------------------------------------
C0A6 A5 FC   MULS     LDA  MR+1
C0A8 10 CB            BPL  MULU     ;MR POSITIV, NICHTS WEITER ZU TUN
C0AA 38               SEC   ;MR NEGIEREN
C0AB A9 00            LDA  #0
C0AD E5 FB            SBC  MR
C0AF 85 FB            STA  MR
C0B1 A9 00            LDA  #0
C0B3 E5 FC            SBC  MR+1
C0B5 85 FC            STA  MR+1
C0B7 20 75 C0         JSR  MULU
C0BA 85 FF            STA  TEMP
C0BC 38               SEC   ;PRODUKT NEGIEREN
C0BD A9 00            LDA  #0
C0BF E5 FF            SBC  TEMP
C0C1 A8               TAY
C0C2 A9 00            LDA  #0
C0C4 E5 FB            SBC  MR
C0C6 85 FB            STA  MR
C0C8 A9 00            LDA  #0
C0CA E5 FC            SBC  MR+1
C0CC 85 FC            STA  MR+1
C0CE 98               TYA
C0CF 60               RTS
;----------------------------------------------------------------
; LFO N UM EINEN SCHRITT WEITERSCHALTEN
; DAS PROGRAMM ERWARTET EINE GROESSE DER GESTALT
; 5 * N  (N=0...6) IN LFONR
;----------------------------------------------------------------
C0D0 A6 FE   LFO      LDX  LFONR
C0D2 BD 24 C0         LDA  LFOC,X
C0D5 29 06            AND  #$06
C0D7 C9 04            CMP  #$04     ;HOLD "?"
C0D9 F0 12            BEQ  LFORTS   ;DANN NICHTS ZU TUN
C0DB C9 06            CMP  #$06     ;RUN "?"
C0DD F0 0F            BEQ  LFORUN
C0DF A9 00   LFORES   LDA  #0       ;RESET
C0E1 9D 49 C0         STA  SAWUP,X  ;DYNAMISCHE PARAMETER
C0E4 9D 4A C0         STA  SAWUP+1,X ;INITIALISIEREN
C0E7 9D 4B C0         STA  KURVE,X
C0EA 9D 4C C0         STA  KURVE+1,X
C0ED 60      LFORTS   RTS
C0EE 18      LFORUN   CLC
C0EF BD 49 C0         LDA  SAWUP,X
C0F2 7D 20 C0         ADC  LFOF,X
C0F5 9D 49 C0         STA  SAWUP,X
C0F8 85 FB            STA  MR
C0FA BD 4A C0         LDA  SAWUP+1,X
C0FD 7D 21 C0         ADC  LFOF+1,X
C100 9D 4A C0         STA  SAWUP+1,X
C103 85 FC            STA  MR+1     ;SAWUP=SAWUP+LFOF
; KURVENFORM ERMITTELN
C105 BD 24 C0         LDA  LFOC,X
C108 29 18            AND  #$18
C10A F0 27            BEQ  TRIAN
C10C C9 08            CMP  #$08
C10E F0 50            BEQ  LFOMUL   ;SAWUP, WENIG ZU TUN
C110 C9 10            CMP  #$10
C112 F0 40            BEQ  SAWDOWN
; SQUARE
; KURVE = + LFOA(8) * 2**7 , FALLS SAWUP < LFOP
; KURVE = - LFOA(8) * 2**7 , SONST
C114 A5 FC            LDA  MR+1
C116 DD 22 C0         CMP  LFOP,X
C119 90 09            BCC  SQPOS
C11B 38      SQNEG    SEC   ;LFOA NEGIEREN
C11C A9 00            LDA  #0
C11E FD 23 C0         SBC  LFOA,X
C121 38               SEC
C122 B0 04            BCS  SQ1      ;(IMMER)
C124 18      SQPOS    CLC
C125 BD 23 C0         LDA  LFOA,X
C128 6A      SQ1      ROR  A        ;ARTITHMETISCHER RECHTS-SHIFT
C129 9D 4C C0         STA  KURVE+1,X
C12C A9 00            LDA  #0
C12E 6A               ROR  A
C12F 9D 4B C0         STA  KURVE,X
C132 60               RTS
; TRIAN
; BERECHNE MAX(SAWUP, -SAWUP-1) * 2 - 2**15
C133 A5 FC   TRIAN    LDA  MR+1
C135 10 13            BPL  TRPOS
C137 A5 FB   TRNEG    LDA  MR
C139 49 FF            EOR  #$FF
C13B 0A               ASL  A
C13C 85 FB            STA  MR
C13E A5 FC            LDA  MR+1
C140 49 FF            EOR  #$FF
C142 2A               ROL  A        ;(-SAWUP-1)*2
C143 49 80            EOR  #$80     ;-2**15
C145 85 FC            STA  MR+1
C147 4C 60 C1         JMP  LFOMUL
C14A 06 FB   TRPOS    ASL  MR
C14C 2A               ROL  A        ;SAWUP*2
C14D 49 80            EOR  #$80     ;-2**15
C14F 85 FC            STA  MR+1
C151 4C 60 C1         JMP  LFOMUL
; SAWDOWN
; BERECHNE -SAWUP-1
C154 A5 FB   SAWDOWN  LDA  MR
C156 49 FF            EOR  #$FF
C158 85 FB            STA  MR
C15A A5 FC            LDA  MR+1
C15C 49 FF            EOR  #$FF
C15E 85 FC            STA  MR+1
; LFOMUL
; ZWEIERKOMPLEMENT-KURVENFORM IN MR(16)
; MIT LFOA(8) MULTIPLIZIEREN
C160 BD 23 C0 LFOMUL  LDA  LFOA,X
C163 85 FD            STA  MD
C165 20 A6 C0         JSR  MULS
C168 A6 FE            LDX  LFONR
C16A A5 FB            LDA  MR
C16C 9D 4B C0         STA  KURVE,X
C16F A5 FC            LDA  MR+1
C171 9D 4C C0         STA  KURVE+1,X
C174 60               RTS
;----------------------------------------------------------------
; EG (ADSR) UM EINEN SCHRITT WEITERSCHALTEN
;----------------------------------------------------------------
C175 AD 48 C0 EG      LDA  EGC
C178 29 06            AND  #$06
C17A C9 04            CMP  #$04     ;HOLD "?"
C17C F0 15            BEQ  EGRTS    ;DANN NICHTS ZU TUN
C17E C9 06            CMP  #$06     ;RUN "?"
C180 F0 12            BEQ  EGRUN
C182 A9 00   EGRES    LDA  #0       ;RESET
C184 8D 6C C0         STA  E        ;DYNAMISCHE PARAMETER
C187 8D 6D C0         STA  E+1      ;INITIALISIEREN
C18A 8D 6E C0         STA  EKURVE
C18D 8D 6F C0         STA  EKURVE+1
C190 8D 70 C0         STA  EPHASE
C193 60      EGRTS    RTS
C194 A9 01   EGRUN    LDA  #1       ;MASKE FUER BIT 0
C196 2C 48 C0         BIT  EGC      ;GATE "?"
C199 F0 52            BEQ  RELEASE
; ATTACK ODER DECAY
C19B 2C 70 C0         BIT  EPHASE
C19E D0 39            BNE  DECAY
C1A0 AD 6C C0 ATTACK  LDA  E
C1A3 49 FF            EOR  #$FF
C1A5 85 FB            STA  MR
C1A7 AD 6D C0         LDA  E+1
C1AA 49 FF            EOR  #$FF
C1AC 85 FC            STA  MR+1     ;MR(16)=2**15-1-E(16)
C1AE AD 43 C0         LDA  A
C1B1 0A               ASL  A        ;*2
C1B2 B0 05            BCS  ATTACK2  ;FALLS A>=128,DANN MR * 1
C1B4 85 FD            STA  MD
C1B6 20 75 C0         JSR  MULU     ;INC(16)=MR(16)*2*A(8)/2**8
C1B9 18      ATTACK2  CLC   ;MR(16)=INC(16)
C1BA AD 6C C0         LDA  E
C1BD 65 FB            ADC  MR
C1BF 8D 6C C0         STA  E
C1C2 85 FB            STA  MR
C1C4 AD 6D C0         LDA  E+1
C1C7 65 FC            ADC  MR+1
C1C9 8D 6D C0         STA  E+1
C1CC 85 FC            STA  MR+1     ;E(16)=E(16)+INC(16)
C1CE C9 FF            CMP  #$FF     ;EG(16)>=$FF00 "?"
C1D0 90 47            BCC  EGMUL    ;<
C1D2 A9 01            LDA  #1
C1D4 8D 70 C0         STA  EPHASE   ;UEBERGANG ZU DECAY
C1D7 D0 40            BNE  EGMUL    ;(IMMER)
C1D9 AD 6C C0 DECAY   LDA  E
C1DC 85 FB            STA  MR
C1DE 38               SEC
C1DF AD 6D C0         LDA  E+1
C1E2 ED 45 C0         SBC  S
C1E5 85 FC            STA  MR+1     ;MR(16)=E(16)-S(8)*2**8
C1E7 AD 44 C0         LDA  D
C1EA 4C FF C1         JMP  DECREL
C1ED A9 00   RELEASE  LDA  #0
C1EF 8D 70 C0         STA  EPHASE   ;BEI GATE=1 WIEDER ATTACK
C1F2 AD 6C C0         LDA  E
C1F5 85 FB            STA  MR
C1F7 AD 6D C0         LDA  E+1
C1FA 85 FC            STA  MR+1     ;MR(16)=E(16)
C1FC AD 46 C0         LDA  R
C1FF 85 FD   DECREL   STA  MD       ;GEMEINSAMER TEIL FUER DECAY UND RELEASE
C201 20 75 C0         JSR  MULU     ;MR(16)=DEC(16)
C204 38               SEC
C205 AD 6C C0         LDA  E
C208 E5 FB            SBC  MR
C20A 8D 6C C0         STA  E
C20D 85 FB            STA  MR
C20F AD 6D C0         LDA  E+1
C212 E5 FC            SBC  MR+1
C214 8D 6D C0         STA  E+1
C217 85 FC            STA  MR+1
C219 AD 47 C0 EGMUL   LDA  EGA      ;HUELLKURVE SKALIEREN
C21C 85 FD            STA  MD
C21E 20 75 C0         JSR  MULU     ;MR(16)=E(16)*EGA(8)/2**8
C221 A9 08            LDA  #$08     ;MASKE FUER BIT 3
C223 2C 48 C0         BIT  EGC      ;+/- "?"
C226 F0 10            BEQ  EGPLUS
C228 38      EGMINUS  SEC   ;EKURVE NEGIEREN
C229 A9 00            LDA  #0
C22B E5 FB            SBC  MR
C22D 8D 6E C0         STA  EKURVE
C230 A9 00            LDA  #0
C232 E5 FC            SBC  MR+1
C234 8D 6F C0         STA  EKURVE+1
C237 60               RTS
C238 A5 FB   EGPLUS   LDA  MR       ;EKURVE(16)=MR(16)
C23A 8D 6E C0         STA  EKURVE
C23D A5 FC            LDA  MR+1
```

```
4080:  C23F 8D 6F C0            STA  EKURVE+1
4090:  C242 60                  RTS
 ;------------------------------------------------------------
 ; SUMMIERE MODULATIONSBEITRAEGE GEMAESS EINER KSV-ZEILE
 ; DAS PROGRAMM ERWARTET EIN KSV-BYTE IN AKKU A
 ; UND LIEFERT MODULATIONS-SUMME IN MR(16) AB
 ;------------------------------------------------------------
4150:  C243 49 FF    SUMMOD     EOR  #$FF      ;KSV-BYTE INVERTIEREN
4160:  C245 85 FF               STA  TEMP
4170:  C247 A9 00               LDA  #0
4180:  C249 85 FB               STA  MR
4190:  C24B 85 FC               STA  MR+1
4200:  C24D A0 08               LDY  #8        ;SCHLEIFENZAEHLER
4210:  C24F 46 FF    SUMLOOP    LSR  TEMP
4220:  C251 B0 11               BCS  SUMNEXT   ;BEI 1 NICHTS SUMMIEREN
4230:  C253 AA                  TAX   ;ADRESSVERSATZ FUER LFO-BLOECKE
4240:  C254 A5 FB               LDA  MR
4250:  C256 7D 4B C0            ADC  KURVE,X
4260:  C259 85 FB               STA  MR
4270:  C25B A5 FC               LDA  MR+1
4280:  C25D 7D 4C C0            ADC  KURVE+1,X
4290:  C260 85 FC               STA  MR+1
4300:  C262 8A                  TXA
4310:  C263 38                  SEC
4320:  C264 69 04    SUMNEXT    ADC  #4        ;+5
4330:  C266 88                  DEY
4340:  C267 D0 E6               BNE  SUMLOOP
4350:  C269 60                  RTS
 ;------------------------------------------------------------
 ; PORTAMENTO-EINZELSCHRITT FUER STIMME N
 ; PROGRAMM ERWARTET 7 * N (N=0,1,2) IN STINR
 ;------------------------------------------------------------
4400:  C26A A6 FE    PORT       LDX  STINR
4410:  C26C BD 04 C0            LDA  PORTA,X
4420:  C26F D0 0D               BNE  PORTRUN
4430:  C271 BD 00 C0            LDA  F,X       ;PORTA=0, FREQUENZ UEBERNEHMEN
4440:  C274 9D 05 C0            STA  FP,X
4450:  C277 BD 01 C0            LDA  F+1,X
4460:  C27A 9D 06 C0            STA  FP+1,X
4470:  C27D 60                  RTS
4480:  C27E BD 01 C0 PORTRUN    LDA  F+1,X
4490:  C281 DD 06 C0            CMP  FP+1,X
4500:  C284 90 3A               BCC  PMINUS    ;F<FP
4510:  C286 D0 0B               BNE  PPLUS     ;F>FP
 ; GLEICHHEIT, LOW-BYTES VERGLEICHEN
4530:  C288 BD 00 C0            LDA  F,X
4540:  C28B DD 05 C0            CMP  FP,X
4550:  C28E 90 30               BCC  PMINUS    ;F<FP
4560:  C290 D0 01               BNE  PPLUS     ;F>FP
4570:  C292 60                  RTS   ;F=FP, NICHTS ZU TUN
4580:  C293 38       PPLUS      SEC
4590:  C294 BD 00 C0            LDA  F,X
4600:  C297 FD 05 C0            SBC  FP,X
4610:  C29A 85 FB               STA  MR
4620:  C29C BD 01 C0            LDA  F+1,X
4630:  C29F FD 06 C0            SBC  FP+1,X
4640:  C2A2 85 FC               STA  MR+1      ;DIF(16)=F(16)-FP(16)
4650:  C2A4 BD 04 C0            LDA  PORTA,X
4660:  C2A7 85 FD               STA  MD
4670:  C2A9 20 75 C0            JSR  MULU      ;INC(16)=DIF(16)*PORTA(8)/2**8
4680:  C2AC A6 FE               LDX  STINR
4690:  C2AE 38                  SEC
4700:  C2AF BD 05 C0            LDA  FP,X
4710:  C2B2 65 FB               ADC  MR
4720:  C2B4 9D 05 C0            STA  FP,X
4730:  C2B7 BD 06 C0            LDA  FP+1,X
4740:  C2BA 65 FC               ADC  MR+1
4750:  C2BC 9D 06 C0            STA  FP+1,X    ;FP(16)=FP(16)+INC(16)+1
4760:  C2BF 60                  RTS
4770:  C2C0 38       PMINUS     SEC
4780:  C2C1 BD 05 C0            LDA  FP,X
4790:  C2C4 FD 00 C0            SBC  F,X
4800:  C2C7 85 FB               STA  MR
4810:  C2C9 BD 06 C0            LDA  FP+1,X
4820:  C2CC FD 01 C0            SBC  F+1,X
4830:  C2CF 85 FC               STA  MR+1      ;DIF(16)=FP(16)-F(16)
4840:  C2D1 BD 04 C0            LDA  PORTA,X
4850:  C2D4 85 FD               STA  MD
4860:  C2D6 20 75 C0            JSR  MULU      ;DEC(16)=DIF(16)*PORTA(8)/2**8
4870:  C2D9 A6 FE               LDX  STINR
4880:  C2DB 18                  CLC
4890:  C2DC BD 05 C0            LDA  FP,X
4900:  C2DF E5 FB               SBC  MR
4910:  C2E1 9D 05 C0            STA  FP,X
4920:  C2E4 BD 06 C0            LDA  FP+1,X
4930:  C2E7 E5 FC               SBC  MR+1
4940:  C2E9 9D 06 C0            STA  FP+1,X    ;FP(16)=FP(16)-DEC(16)-1
4950:  C2EC 60                  RTS
 ;------------------------------------------------------------
 ; HAUPTPROGRAMM
 ; SCHALTE ALLE MODULATINSQUELLEN UM EINEN SCHRITT WEITER
 ; MODULIERE ALLE PARAMETER GEMAESS KREUZSCHIENENVERTEILER (KSV)
 ;------------------------------------------------------------
 ; 7 LFOS WEITERSCHALTEN
5020:  C2ED A9 1E    MODUL      LDA  #30
5030:  C2EF 85 FE    LFOLOOP    STA  LFONR
5040:  C2F1 20 D0 C0            JSR  LFO
5050:  C2F4 38                  SEC
5060:  C2F5 A5 FE               LDA  LFONR
5070:  C2F7 E9 05               SBC  #5
5080:  C2F9 10 F4               BPL  LFOLOOP
 ; EG (ADSR) WEITERSCHALTEN
5100:  C2FB 20 75 C1            JSR  EG
 ;
 ; 3 STIMMEN BEARBEITEN
 ;
5140:  C2FE A9 02               LDA  #2
5150:  C300 A2 0E               LDX  #14
5160:  C302 85 9B               STA  ZAEHLER
5170:  C304 86 FE               STX  STINR
 ; FREQUENZ MODULIEREN
5190:  C306 20 6A C2 FMOD       JSR  PORT      ;FP WEITERSCHALTEN
5200:  C309 A6 9B               LDX  ZAEHLER
5210:  C30B BD 18 C0            LDA  KSV,X
5220:  C30E D0 11               BNE  FMOD1
 ; KEINE FREQUENZMODULATION, PARAMETER UEBERNEHMEN
5240:  C310 A6 FE               LDX  STINR
5250:  C312 BD 05 C0            LDA  FP,X
5260:  C315 9D 00 D4            STA  SID,X
5270:  C318 BD 06 C0            LDA  FP+1,X
5280:  C31B 9D 01 D4            STA  SID+1,X
5290:  C31E 4C 41 C3            JMP  PMOD
5300:  C321 20 43 C2 FMOD1      JSR  SUMMOD    ;LIEFERT MODULATIONSWERT IN MR(16)
5310:  C324 A6 FE               LDX  STINR
5320:  C326 BD 06 C0            LDA  FP+1,X
5330:  C329 85 FD               STA  MD
5340:  C32B 20 A6 C0            JSR  MULS      ;MODULATIONSWERT MIT FP HIGH SKALIEREN
5350:  C32E A6 FE               LDX  STINR
5360:  C330 18                  CLC
5370:  C331 BD 05 C0            LDA  FP,X
5380:  C334 65 FB               ADC  MR
5390:  C336 9D 00 D4            STA  SID,X
5400:  C339 BD 06 C0            LDA  FP+1,X
5410:  C33C 65 FC               ADC  MR+1
5420:  C33E 9D 01 D4            STA  SID+1,X
 ; PULSWEITE MODULIEREN
5440:  C341 A6 9B    PMOD       LDX  ZAEHLER
5450:  C343 BD 1B C0            LDA  KSV+3,X
5460:  C346 D0 11               BNE  PMOD1
 ; KEINE PW-MODULATION, PARAMETER UEBERNEHMEN
5480:  C348 A6 FE               LDX  STINR
5490:  C34A BD 02 C0            LDA  PW,X
5500:  C34D 9D 02 D4            STA  SID+2,X
5510:  C350 BD 03 C0            LDA  PW+1,X
5520:  C353 9D 03 D4            STA  SID+3,X
5530:  C356 4C 81 C3            JMP  NEXTSTI
5540:  C359 20 43 C2 PMOD1      JSR  SUMMOD    ;LIEFERT MODULATIONSWERT NACH MR(16)
 ; MR(12)=MR(16)/2**4, ERGIBT 12-BIT ZWEIERKOMPLEMENTGROESSE
5560:  C35C 46 FC               LSR  MR+1
5570:  C35E 66 FB               ROR  MR
5580:  C360 46 FC               LSR  MR+1
5590:  C362 66 FB               ROR  MR
5600:  C364 46 FC               LSR  MR+1
5610:  C366 66 FB               ROR  MR
5620:  C368 46 FC               LSR  MR+1
5630:  C36A 66 FB               ROR  MR
5640:  C36C A6 FE               LDX  STINR
5650:  C36E 18                  CLC
5660:  C36F BD 02 C0            LDA  PW,X
5670:  C372 65 FB               ADC  MR
5680:  C374 9D 02 D4            STA  SID+2,X
5690:  C377 BD 03 C0            LDA  PW+1,X
5700:  C37A 65 FC               ADC  MR+1
5710:  C37C 29 0F               AND  #$0F      ;BIT 7-4 AUSBLENDEN
5720:  C37E 9D 03 D4            STA  SID+3,X
5730:  C381 8A       NEXTSTI    TXA   ;(STINR)
5740:  C382 38                  SEC
5750:  C383 E9 07               SBC  #7
5760:  C385 85 FE               STA  STINR
5770:  C387 C6 9B               DEC  ZAEHLER
5780:  C389 30 03               BMI  FILMOD
5790:  C38B 4C 06 C3            JMP  FMOD      ;NAECHSTE STIMME
 ; FILTERFREQUENZ MODULIEREN, NUR HIGH-BYTE
5810:  C38E AD 15 C0 FILMOD     LDA  FILT      ;LOW-BYTE UEBERNEHMEN
5820:  C391 8D 15 D4            STA  SID+21
5830:  C394 AD 1E C0            LDA  KSV+6
5840:  C397 D0 09               BNE  FILMOD1
 ; KEINE FILTERMODULATION, PARAMETER UEBERNEHMEN
5860:  C399 AD 16 C0            LDA  FILT+1
5870:  C39C 8D 16 D4            STA  SID+22
5880:  C39F 4C AE C3            JMP  LAUTMOD
5890:  C3A2 20 43 C2 FILMOD1    JSR  SUMMOD    ;LIEFERT MODULATIONSWERT IN MR(16)
5900:  C3A5 18                  CLC
5910:  C3A6 AD 16 C0            LDA  FILT+1
5920:  C3A9 65 FC               ADC  MR+1
5930:  C3AB 8D 16 D4            STA  SID+22
 ; LAUTSTAERKE MODULIEREN
 ; NUR DIE 4 OBEREN BITS VON MR+1 TRAGEN DAZU BEI
5960:  C3AE AD 1F C0 LAUTMOD    LDA  KSV+7
5970:  C3B1 D0 07               BNE  LAUMOD1
 ; KEINE LAUTSTAERKEMODULATION, PARAMETER UEBERNEHMEN
5990:  C3B3 AD 17 C0            LDA  MODLAUT
6000:  C3B6 8D 18 D4            STA  SID+24
6010:  C3B9 60                  RTS
6020:  C3BA 20 43 C2 LAUMOD1    JSR  SUMMOD    ;LIEFERT MODULATIONSWERT IN MR(16)
6030:  C3BD AD 17 C0            LDA  MODLAUT
6040:  C3C0 29 F0               AND  #$F0      ;MODUS (BIT 7-4) EXTRAHIEREN
6050:  C3C2 85 FF               STA  TEMP
6060:  C3C4 A5 FC               LDA  MR+1
6070:  C3C6 4A                  LSR  A
6080:  C3C7 4A                  LSR  A
6090:  C3C8 4A                  LSR  A
6100:  C3C9 4A                  LSR  A         ;A(4)=MR+1(8)/2**4
6110:  C3CA 18                  CLC
6120:  C3CB 6D 17 C0            ADC  MODLAUT
6130:  C3CE 29 0F               AND  #$0F      ;BIT 7-4 AUSBLENDEN
6140:  C3D0 05 FF               ORA  TEMP      ;MODUS EINBLENDEN
6150:  C3D2 8D 18 D4            STA  SID+24
6160:  C3D5 60                  RTS
 ;------------------------------------------------------------
 ; CIA#1 TIMER A ABFRAGEN, LOW-BYTE IN A, HIGH-BYTE IN X
 ;------------------------------------------------------------
6200:  C3D6 AD 04 DC TIME       LDA  $DC04     ;TIMER A LOW
6210:  C3D9 AE 05 DC            LDX  $DC05     ;TIMER A HIGH
6220:  C3DC C9 04               CMP  #4
6230:  C3DE B0 01               BCS  TIME1
 ; TA LOW < 4, UNTERLAUF NACH TA HIGH KORRIGIEREN
6250:  C3E0 E8                  INX
6260:  C3E1 60       TIME1      RTS
 ;------------------------------------------------------------
 ; ERWEITERTES INTERUPTPROGRAMM
 ;------------------------------------------------------------
6300:  C3E2 20 D6 C3 INTRPT     JSR  TIME      ;STARTZEIT LESEN
6310:  C3E5 8D 73 C0            STA  ZEIT1     ;UND FESTHALTEN
6320:  C3E8 8E 74 C0            STX  ZEIT1+1
6330:  C3EB 20 ED C2            JSR  MODUL     ;MODULATIONSSCHRITT
6340:  C3EE 20 D6 C3            JSR  TIME      ;ENDZEIT LESEN
6350:  C3F1 85 FF               STA  TEMP      ;DIFFERENZ BERECHNEN
6360:  C3F3 38                  SEC
6370:  C3F4 AD 73 C0            LDA  ZEIT1
6380:  C3F7 E5 FF               SBC  TEMP
6390:  C3F9 8D 71 C0            STA  ZEIT
6400:  C3FC 86 FF               STX  TEMP
6410:  C3FE AD 74 C0            LDA  ZEIT1+1
6420:  C401 E5 FF               SBC  TEMP
6430:  C403 8D 72 C0            STA  ZEIT+1
6440:  C406 4C 31 EA            JMP  $EA31     ;KERNAL-SYSTEMINTERUPTROUTINE
 ;------------------------------------------------------------
 ; INTERUPTVEKTOR UMSTELLEN (MODULATOR EINSCHALTEN)
 ;------------------------------------------------------------
6480:  C409 78       START      SEI
6490:  C40A A9 E2               LDA  #<INTRPT
6500:  C40C 8D 14 03            STA  $0314
6510:  C40F A9 C3               LDA  #>INTRPT
6520:  C411 8D 15 03            STA  $0315
6530:  C414 58                  CLI
6540:  C415 60                  RTS
 ;------------------------------------------------------------
 ; INTERUPTVEKTOR ZURUECKSTELLEN (MODULATOR AUSSCHALTEN)
 ;------------------------------------------------------------
6580:  C416 78       AUS        SEI
6590:  C417 A9 31               LDA  #$31
6600:  C419 8D 14 03            STA  $0314
6610:  C41C A9 EA               LDA  #$EA
6620:  C41E 8D 15 03            STA  $0315
6630:  C421 58                  CLI
6640:  C422 60                  RTS
0C075-C423

READY.
```

Dokumentiertes Assemblerlisting von »Modulator« (Schluß)


64ER ONLINE

Effektives Programmieren (5)

Sortieren in Basic — Teil 2

Einfache Sortieralgorithmen sind leider auch die langsamsten. Dennoch lassen sie sich durch einige kleinere Änderungen noch erheblich verbessern, so zum Beispiel Bubblesort. Wesentlich komplizierter ist da schon Shellsort, dafür aber auch schneller. Wir zeigen Ihnen, wie es funktioniert.

```
10000 REM SORTIEREN DURCH AUSTAUSCHEN          <059>
10010 REM VERBESSERT                           <213>
10020 REM                                      <218>
10030 REM BUBBLESORT 2                         <010>
10032 REM G IST DIE LETZTE POSITION BEIM       <089>
10034 REM VERTAUSCHEN                          <048>
10036 REM F ZEIGT VERTAUSCHUNG AN              <225>
10040 G=A-1:FOR X=A-1 TO 1 STEP-1              <177>
10050 F=0:FOR Y=1 TO G                         <115>
10060 IF A$(Y)<=A$(Y+1)THEN 10080              <252>
10065 REM AUSTAUSCHEN BEIDER ELEMENTE          <069>
10070 F=Y:S$=A$(Y):A$(Y)=A$(Y+1):A$(Y+1)=S
      $                                        <114>
10080 NEXT Y                                   <098>
10090 G=F:IF F=0 THEN 10120                    <088>
10100 GOSUB 3000: REM AUSGABE                  <172>
10110 NEXT X                                   <127>
10120 REM ENDE                                 <090>
```

Listing 1. Der verbesserte Bubblesort-Algorithmus

```
SPU IOF CEH FSO AIF XKY BHW QTR OPC KBL
IOF CEH FSO AIF SPU BHW QTR OPC KBL XKY
CEH FSO AIF IOF BHW QTR OPC KBL SPU XKY
CEH AIF FSO BHW IOF OPC KBL QTR SPU XKY
AIF CEH BHW FSO IOF KBL OPC QTR SPU XKY
AIF BHW CEH FSO IOF KBL OPC QTR SPU XKY

AIF BHW CEH FSO IOF KBL OPC QTR SPU XKY
 10  ELEMENTE
```

Bild 1. Die Wirkung von Bubblesort 2. Durch kleine Änderungen wird Bubblesort um einiges schneller. Die unterstrichenen Werte wurden an den richtigen Platz gesetzt;

In der letzten Folge beschäftigten wir uns mit straight insertion und mit Bubblesort, zwei sehr einfachen Sortieralgorithmen. Diesmal wollen wir das Niveau schon ein wenig anheben, um uns dem eigentlichen Ziel unseres Kurses langsam zu nähern. Letztendlich geht es uns nur darum, eine möglichst schnelle und effektive Sortiermethode für praktische Anwendungen zu suchen. Fangen wir deshalb gleich einmal mit der Verbesserung eines Sortieralgorithmus an, der letztes Mal besprochen wurde.

Haben Sie sich mit Bubblesort schon intensiver beschäftigt? Wenn ja, werden Sie auch ganz bestimmt dessen Schwächen ausfindig gemacht haben. Wir erinnern uns: Bubblesort fängt am Anfang eines Variablenfeldes an und vergleicht die beiden ersten Variablen. Steht die größere der beiden weiter vorne, so werden die Variablen vertauscht. Jetzt vergleicht er die zweite mit der dritten Variablen des Arrays und setzt dieses Vergleichen und Austauschen solange fort, bis das gesamte Feld durchgearbeitet ist und die größte Variable jetzt am Ende des Arrays steht. Als nächstes wird das Variablenfeld um die letzte Variable vermindert, so daß jetzt der zweitgrößte String auf die gleiche Art und Weise »nach unten« befördert wird. Diese Vorgänge wiederholen sich so lange, bis nur noch eine Variable übrigbleibt, die jetzt die kleinste ist.

Bubblesort optimiert

Nun aber zu den Schwächen von Bubblesort. Ist Ihnen beim Ausprobieren des Programms aus der letzten Folge vielleicht aufgefallen, daß Bubblesort sehr »stur« arbeitet? Es kann nämlich ohne weiteres passieren, daß ein Feld bereits nach dem dritten Durchgang vollständig sortiert vorliegt. Dies wird von Bubblesort jedoch nicht erkannt. Der Computer »sortiert« weiter, bis alle Durchläufe erledigt sind.

Dieses Problem können wir ganz einfach lösen, indem wir ein Flag einsetzen, das uns anzeigt, ob im letzten Durchgang noch eine Vertauschung stattgefunden hat. Wurde kein Tausch mehr vorgenommen, so wird der Sortiervorgang beendet. Dieses Flag ist schon eine ziemliche Verbesserung gegenüber der Rohversion, aber wir wollen uns damit noch nicht zufriedengeben.

Es kann beim Sortieren auch durchaus der Fall eintreten, daß im letzten Durchlauf nur noch beispielsweise drei Vertauschungen im ersten Drittel des Feldes stattgefunden haben. Die letzten beiden Drittel des Feldes sind also bereits sortiert.

Damit Bubblesort auch diesen Fall erkennt, wird eine zweite zusätzliche Variable eingeführt, die die Position der jeweils letzten Vertauschung eines Durchlaufes beinhaltet. Es wird nun im weiteren Verlauf immer nur bis zu dieser Position gearbeitet, da der Rest des Feldes bereits sortiert vorliegen muß.

Mit diesen beiden Verbesserungen wollen wir es aber bereits gut sein lassen (Listing 1, Bild 1). Der neue Bubblesort-Algorithmus arbeitet besonders bei schon teilsortierten Feldern ziemlich effizient; ist der »alten« Version jedoch bei total vermischten Feldern infolge der zusätzlichen (Zeit verbrauchenden) »Erweiterungen« unterlegen.

Bubblesort soll uns nun nicht weiter beschäftigen, denn trotz seines wohlklingenden Namens ist er so ziemlich der langsamste Algorithmus, den es gibt.

An dieser Stelle gleich einmal ein paar Bemerkungen zur Zeitmessung: Die jetzt vorgestellten Algorithmen, die Sie jeweils als Listings abgedruckt finden, sind in der Form zur Zeitmessung natürlich nicht geeignet. Das liegt daran, daß die Programme so aufgebaut sind, daß Sie den Algorithmus leicht nachvollziehen können, was natürlich auf Kosten der Geschwindigkeit geht und die Ergebnisse verfälschen würde.

Im abschließenden Artikel über die Sortiermethoden werden wir die einzelnen Programme jedoch auch unter dem Aspekt »Zeit« einander gegenüberstellen. Hier werden wir auch auf das Problem der Garbage Collection eingehen, die uns beim Sortieren von größeren Feldern, je nach Algorithmus, ganz schön in Schwierigkeiten bringen kann, wenn es um eine Zeitmessung geht.

Ein weiteres Problem bei der Zeitmessung ist aber auch die Eigenart der einzelnen Sortiermethoden. Ich erwähnte schon in der letzten Folge, daß es natürliche und unnatürliche Algorithmen gibt, wobei die natürlichen dann am schnellsten arbeiten, wenn das Feld schon sortiert vorliegt.

Für die Mathematiker unter Ihnen ist jedem Sortieralgorithmus eine kleine Formel zur Berechnung der mittleren (!) Sortierzeit beigefügt. Diese Formel dient nur der Gesamtbetrachtung und zeigt jeweils, warum die einen Algorithmen so langsam und andere wesentlich schneller sind.

straight selection

Nun aber zu einer neuen Sortiermethode. Es handelt sich hierbei um ein Sortieren durch direktes Auswählen, was durch einen englischen Ausdruck wieder passend beschrieben wird: straight selection.

Auch straight selection ist ein relativ einfacher Algorithmus, dessen Funktionsweise wir uns gleich etwas näher betrachten wollen (Bild 2).

Im ersten Durchgang sucht der Computer nach dem größten Element im Feld. Wird dieses gefunden, so erfolgt eine Vertauschung zwischen diesem Element und dem allerletzten des Feldes, da die größte Variable logischerweise am Schluß stehen muß. Jetzt wird die Länge des Feldes durch Wegnahme des letzten Elements um 1 vermindert. Danach wird in diesem »Rest-Array« wiederum nach dem größten Element gesucht und dieses ebenfalls mit dem


64ER ONLINE

letzten Element (das jetzt das vorletzte des Gesamtfeldes ist) vertauscht. Dieser Vorgang wiederholt sich so lange, bis die Länge des Restfeldes 1 ist und wir an erster Position zwangsläufig das kleinste Element erhalten.

In Bild 3 können Sie die Arbeitsweise von straight selection an einem praktischen Beispiel nachvollziehen, wobei immer jene Elemente unterstrichen sind, die im nächsten Schritt einsortiert werden.

Natürlich funktioniert straight selection auch andersherum, das heißt Sie können jeweils nach dem kleinsten Element suchen und dieses dann mit dem an erster Stelle stehenden Element vertauschen.

Um Ihnen auch die Zeitverhältnisse zu beschreiben, oder um Ihren mathematischen Geist zu beflügeln (wie Sie wollen), seien an dieser Stelle einmal wieder zwei Formeln über straight selection aufgestellt.

Für seine Arbeit benötigt straight selection eine mittlere Anzahl von Vergleichen, die in etwa durch die folgende Formel angenähert werden, wenn wir davon ausgehen, daß a die Anzahl der zu sortierenden Elemente enthält:

Anzahl Vergleiche: $\frac{a^2 - a}{2}$

Für die Anzahl der Bewegungen innerhalb der Arrays gilt folgende Beschreibung:

Anzahl Bewegungen: a - 1

Mit straight selection haben wir unter anderem gleich das erste Beispiel für einen unnatürlichen Sortieralgorithmus. Wenn wir ein Feld bearbeiten wollen, das schon sortiert vorliegt, so braucht unser Programm sehr lange, um das größte Element ausfindig zu machen, da wir von vorne mit dem Suchen beginnen. Bearbeiten Sie also meistens schon teilsortierte Felder, so ist es ratsam, mit der Suche des größten Elements von hinten zu beginnen. Die Umstellung des Programms in Listing 2 dürfte Ihnen keine Schwierigkeiten bereiten, da lediglich die Suchschleife umzudrehen und mit STEP1 zu versehen ist.

So, das wäre auch schon alles, was zu straight selection zu sagen ist. Wie Sie sehen, ist das immer noch ein sehr einfacher Algorithmus, der in etwa mit straight insertion gleichzusetzen ist, was die Effektivität betrifft. Diese Gleichsetzung gilt aber natürlich nur für zufallsbesetzte Felder.

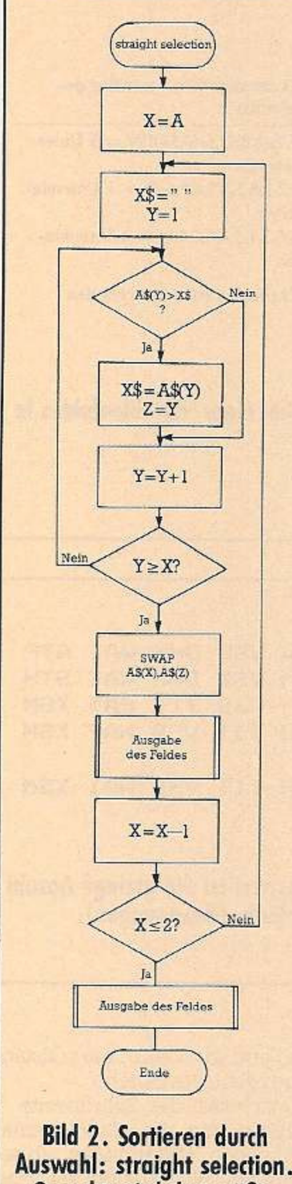


Bild 2. Sortieren durch Auswahl: straight selection. Gesucht wird das größte Element und an das Ende des Feldes gesetzt. Nach jedem Durchgang wird das Feld um ein Element kürzer.

```
10000 REM SORTIEREN DURCH DIREKTES          <082>
10010 REM AUSWAEHLEN                        <189>
10020 REM                                   <218>
10030 REM STRAIGHT SELECTION                <240>
10040 FOR X=A TO 2 STEP-1:X$=""             <050>
10050 FOR Y=1 TO X                          <034>
10060 IF A$(Y)>X$THEN X$=A$(Y):Z=Y          <189>
10070 NEXT Y                                <088>
10080 S$=A$(X):A$(X)=A$(Z):A$(Z)=S$         <059>
10090 GOSUB 3000                            <225>
10100 NEXT X                                <117>
10110 REM ENDE                              <080>
```

Listing 2. Sortieren durch direktes Auswählen: straight selection

```
CSB ONN CSX XDO KXF DVK JJD UWK HVG SCX
CSB ONN CSX SCX KXF DVK JJD UWK HVG XDO
CSB ONN CSX SCX KXF DVK JJD HVG UWK XDO
CSB ONN CSX HVG KXF DVK JJD SCX UWK XDO
CSB JJD CSX HVG KXF DVK ONN SCX UWK XDO
CSB JJD CSX HVG DVK KXF ONN SCX UWK XDO
CSB DVK CSX HVG JJD KXF ONN SCX UWK XDO
CSB DVK CSX HVG JJD KXF ONN SCX UWK XDO
CSB CSX DVK HVG JJD KXF ONN SCX UWK XDO
CSB CSX DVK HVG JJD KXF ONN SCX UWK XDO

CSB CSX DVK HVG JJD KXF ONN SCX UWK XDO
 10  ELEMENTE
```

Bild 3. Straight selection bei der Arbeit. Die jeweils neu einzuordnenden Elemente sind unterstrichen.

```
10000 REM SORTIEREN MIT ABNEHMENDER                   <132>
10010 REM SCHRITTWEITE                                <111>
10020 REM                                             <218>
10030 REM SHELLSORT                                   <164>
10035 DIM AA$(A)                                      <024>
10040 S=INT(A/2): REM SCHRITTWEITE                    <242>
10050 FOR X=1 TO S                                    <028>
10060 FOR Y=1 TO INT(A/S)                             <027>
10070 AA$(Y)=A$((Y-1)*S+X)                            <188>
10080 NEXT Y                                          <098>
10090 AA=Y-1:GOSUB 20000                              <179>
10100 FOR Y=1 TO INT(A/S)                             <067>
10110 A$((Y-1)*S+X)=AA$(Y)                            <228>
10120 NEXT Y                                          <138>
10130 NEXT X                                          <147>
10140 S=INT(S/2)                                      <000>
10150 GOSUB 3000                                      <029>
10160 IF S GOTO 10050                                 <052>
10170 REM ENDE                                        <140>
10180 GOTO 50000                                      <105>
20000 FOR XX=2 TO AA                                  <137>
20010 IF AA$(XX)>=AA$(XX-1) THEN 20080                <049>
20020 REM EINFUEGEN DES ELEMENTS                      <224>
20030 XX$=AA$(XX): FOR YY=XX-1 TO 1 STEP-1            <190>
20040 AA$(YY+1)=AA$(YY)                               <117>
20050 IF XX$<=AA$(YY-1) THEN 20070                    <137>
20060 AA$(YY)=XX$: GOTO 20080                         <150>
20070 NEXT YY                                         <232>
20080 NEXT XX                                         <240>
20090 RETURN                                          <086>
```

Listing 3. Der Shellsort-Algorithmus. Ab Zeile 20000 wird die straight-insertion-Methode benutzt, um die Untereinheiten zu sortieren. Auch hier näheres im Artikel.

Shellsort

Der nächste Sortieralgorithmus trägt den Namen seines Erfinders (D.L.Shell) und wurde 1959 entwickelt. Es handelt sich hierbei schon um einen komplizierteren Algorithmus, den wir deshalb sehr ausführlich besprechen wollen (Bild 4). Shellsort ist ein Sortieren durch direktes Einfügen und gehört damit der gleichen »Familie« wie straight insertion an.

Durch entsprechende Berechnungen hatte Shell herausgefunden, daß sich Sortiervorgänge beschleunigen lassen, wenn nicht nur benachbarte Elemente miteinander verglichen werden, sondern auch weiter voneinander entfernte. Wir vergleichen also beispielsweise nicht mehr das erste Element mit dem zweiten, sondern vielmehr das erste mit dem fünften.

Durch diese Methode erreicht man eine gewisse »Grobsortierung«, die sich jedoch gleichmäßig über das gesamte Feld verteilt. Das so neu entstandene Variablenfeld wird wiederum sortiert, wobei jetzt aber das erste mit dem dritten Element verglichen wird. Die Sortierung wird also durch abnehmende Abstände zunehmend »feiner«, bis beim Abstand 1 die letzte, absolute Sortierung erfolgt.

Unklar? Keine Angst, wir wer-

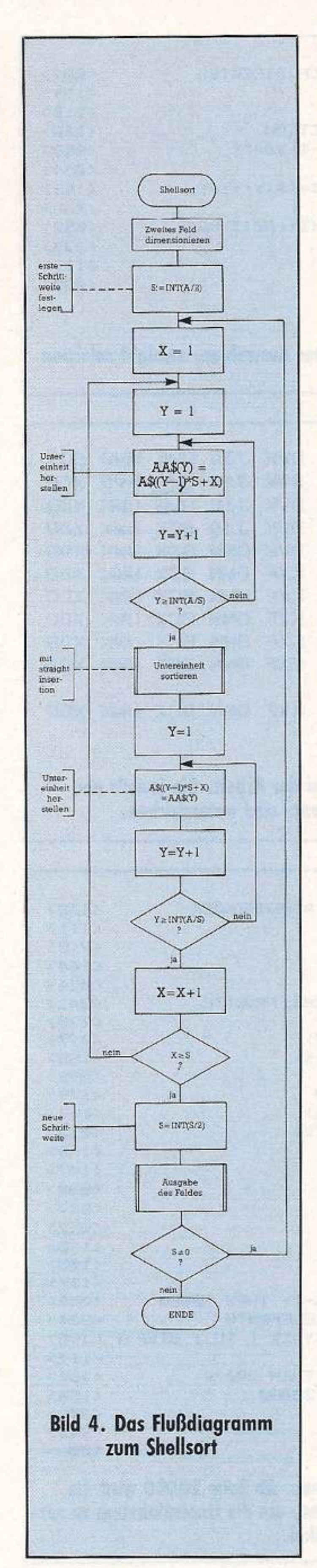


Bild 4. Das Flußdiagramm zum Shellsort

9	1	3	0	4	7	8	6	5	2	:Ausgangsfeld (zufällig gemischt)
9	1	3	0	4	7	8	6	5	2	(9,7)(1,8)(3,6)(0,5)(4,2) = 5 Untereinheiten
7	1	3	0	2	9	8	6	5	4	(7,3,2,8,5) (1,0,9,6,4) = 2 Untereinheiten
2	0	3	1	5	4	7	6	8	9	(2,0,3,1,5,4,7,6,8,9) = 1 Untereinheit
0	1	2	3	4	5	6	7	8	9	:Ergebnis nach 3 Durchläufen

Bild 5. Das Anlegen von Untereinheiten eines Variablenfeldes in Shellsort. Näheres dazu im Artikel.

```
XSM TIT NCQ NDP STH PUW VSB ONM WAI ATP
PUW TIT NCQ NDP ATP XSM VSB ONM WAI STH
ATP NDP NCQ ONM PUW STH VSB TIT WAI XSM
ATP NCQ NDP ONM PUW STH TIT VSB WAI XSM

ATP NCQ NDP ONM PUW STH TIT VSB WAI XSM
  10 ELEMENTE
```

Bild 6. Shellsort in Aktion. Bemerkenswert ist die geringe Anzahl der zum Sortieren notwendigen Durchläufe (Bewegungen).

den das gleich einmal an einem praktischen Beispiel erläutern.

Sehen Sie sich Bild 5 an. Hier haben wir ein zufällig geordnetes Feld mit zehn Elementen. Als ersten Abstandswert nimmt Shellsort üblicherweise a/2, also die Hälfte der Gesamtanzahl der Elemente. In unserem Fall ist das 5.

Aus diesem umsortierten Feld holen wir jetzt alle Zahlen zu Untereinheiten zusammen, die den Abstand (besser: die Schrittweite) 5 haben. In Bild 5 sehen Sie diese Zusammenstellungen: Es wurde also jeweils das 1. mit dem 6., das 2. mit dem 7., das 3. mit dem 8., das 4. mit dem 9. und das 5. mit dem 10. Element zu einer Einheit zusammengefaßt.

Da die Schrittweite 5 ist, kann jede Untereinheit verständlicherweise nur zwei Elemente enthalten. Nun, was sollen wir jetzt mit diesen Untereinheiten machen?

Diese werden sortiert, und zwar verwenden wir dabei einen einfachen und unkomplizierten Sortieralgorithmus, wie zum Beispiel straight insertion.

Wir sortieren also die erste Untereinheit, aus (9,7) wird (7,9). Jetzt schreiben wir diese Untereinheit wieder an die gleiche Position in unser Feld zurück, wobei jedoch die 7 dort steht, wo vorher die 9 stand und umgekehrt. Dann sortieren wir die zweite Untereinheit und schreiben sie ebenso zurück. Das geschieht so lange, bis alle Untereinheiten abgearbeitet worden sind und wir wieder ein vollständiges Array erhalten.

Jetzt wird die Schrittweite 5 halbiert und die Nachkommastelle des Ergebnisses abgeschnitten. Wir erhalten als neue Schrittweite 2. Wieder legen wir uns Untereinheiten an, wobei wir jedoch nur mehr zwei Untereinheiten zu je fünf Elementen bekommen. Wichtig für die Programmentwicklung ist an dieser Stelle die Entdeckung, daß die Anzahl der Untereinheiten grundsätzlich der Schrittweite entspricht.

Auch hier wird mit den Untereinheiten wieder verfahren, wie oben. Sie werden sortiert und wieder in das ursprüngliche Array zurückgeschrieben. Das Ergebnis des letzten Durchlaufes können Sie wieder in Bild 5 ablesen. Der nächste Durchlauf ist schon der letzte; hier ist die Schrittweite nunmehr 1 und es erfolgt eine Schlußsortierung des gesamten Feldes.

Daß Shellsort so schnell ist, obwohl er einige vollständige Sortierläufe als Unterprogramme verwendet, liegt daran, daß das Sortierunterprogramm jeweils ziemlich optimierte Einheiten zur Bearbeitung bekommt. Auch beim letzten Durchgang, wo ja nochmals das gesamte Feld durchsortiert wird, sind die Elemente schon so angeordnet, daß eine Sortierung ohne viele Bewegungen möglich ist. Listing 3 enthält die Shellsortroutine, wobei als Unterprogramm ab Zeile 20000 straight insertion verwendet wird. Sie können einmal verschiedene Algorithmen in Shellsort verwenden; vielleicht finden Sie eine optimale Zusammenstellung? Das Unterprogramm bearbeitet das Array AA$(x) und erwartet die Anzahl der Elemente in AA.

Wenn Sie sich einmal den Beispielausdruck zu Shellsort betrachten (Bild 6), so werden Sie feststellen, daß dieser Algorithmus nur mehr drei Durchgänge für zehn Elemente benötigt. Diese Zahl läßt auf ein gutes Ergebnis hoffen. In der Tat haben wir mit Shellsort schon ein sehr gutes Sortierprogramm, das vielen praktischen Anwendungen gewachsen sein dürfte. Gegenüber der vorher besprochenen Sortieralgorithmen arbeitet Shellsort um einiges schneller, was besonders bei größeren Feldern angenehm auffällt. Für die Schrittweite können übrigens auch andere abfallende Reihen verwendet werden, die mit 1 aufhören. Es hat sich nämlich gezeigt, daß die Wahl der richtigen Reihe entscheidend zur Geschwindigkeit von Shellsort beiträgt.

Wollen wir zu Shellsort eine mathematische Berechnung liefern, wird's schwierig. Dieser Algorithmus ist bereits dermaßen komplex, daß eine Berechnung fast unmöglich wird. Es kann an dieser Stelle nur eine Aussage über die mittlere Sortierzeit gemacht werden, die sich in etwa im Bereich um $a^{1.2}$ bewegt, wobei a wiederum die Anzahl der zu sortierenden Elemente darstellt.

So, mit Shellsort haben wir uns nun endgültig von den einfachen Sortieralgorithmen losgesagt. Wie Sie sehen, kann eine höhere Komplexität der Programme und ein damit verbundener größerer Zeitbedarf, ohne weiteres die Nachteile von einfacheren Programmen aufwiegen. Aber auch hier kommt es natürlich auf die Art der Aufgabenstellung an. Shellsort verträgt zum Beispiel keine umgekehrt sortierten Arrays. Hier wird auch dieser schnelle Sortieralgorithmus langsam.

In der nächsten Folge wollen wir uns ausschließlich mit einem einzigen Sortierprogramm beschäftigen. Es handelt sich um Heapsort. Dieser Algorithmus arbeitet nach dem »Baumprinzip« und ist sehr kompliziert. Aus diesem Grund wollen wir uns ausführlich mit ihm beschäftigen, denn wir haben es dann mit einem der schnellsten Algorithmen zu tun, den es gibt.

(Karsten Schramm/gk)


64ER ONLINE

Funktionen für Anfänger

Auch in Basic kann man Befehle selber entwickeln, zumindest einen bestimmten Typ von Befehl. Und dazu braucht man keine Maschinensprache und keinen Assembler, sondern nur den gesunden Menschenverstand, wie man ihn auch sonst beim Programmieren einsetzt. Gemeint sind die »benutzerdefinierten Funktionen«.

Anfänger haben mit der Definition von neuen Befehlen oft Schwierigkeiten. Das liegt aber nicht an den Anfängern, sondern eher an den meist recht verwirrenden Erklärungen der Handbücher. Handbücher sind zumeist von Computerexperten geschrieben, die oft vergessen, daß ihre Leser erst noch Experten werden wollen und deshalb zunächst mit Begriffen wie »Dummy-Variable« oder »Übergabeparameter« und was es sonst noch an stolzen Termini gibt, nichts anfangen können. Ich jedenfalls konnte es nicht und habe deshalb lange gebraucht, bis ich selbstgestrickte Funktionen so selbstverständlich in meinen Programmen benutzte wie zum Beispiel PRINT.

Vergessen Sie also alles, was Sie bisher verwirrt haben mag, und fangen Sie, zusammen mit mir, noch einmal von vorne zu denken an.

Ich will in drei Schritten vorgehen. Wir wollen zunächst klären, was Funktionen überhaupt sind, was für Eigenschaften sie haben, was sie tun, wofür man sie braucht. Auf dem Hintergrund dieser allgemeineren Informationen wollen wir uns in einem zweiten Schritt der Herstellung eigener Funktionen widmen. Ein drittes Kapitel soll dann ein paar speziellere Hinweise geben. Ein kleiner Anhang schließlich wird ein paar einfache Funktionen zusammenstellen, die nicht die Welt bewegen, sondern nur Sie anregen sollen, sich eine eigene Funktionen-Bibliothek aufzubauen.

1. Funktionen in Basic

1.1 Was ist eine Funktion?

Eine Funktion ist ein Befehl, der den Computer anweist, eine Zahl oder einen Text (Zeichenkette, »String«) zu erzeugen. Die RND-Funktion zum Beispiel erzeugt eine Zufallszahl; die Funktion LEFT$ erzeugt eine Zeichenkette.

Es gibt zwei Grundtypen von Funktionen: solche, die lediglich Daten (Zahlen oder Zeichenketten) ausgeben, und solche, denen man Daten (Zahlen oder Zeichenketten) eingibt, die sie dann in anderer Form wieder ausgeben. Wir wollen die einen Ausgabe-Funktionen nennen und die anderen Eingabe-Ausgabe-Funktionen (Bild 1).

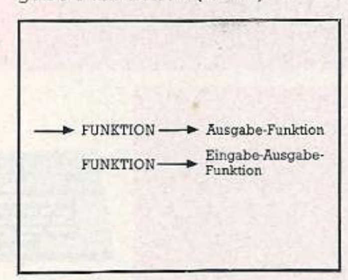


Bild 1. Die zwei Grundtypen von Funktionen

Die oben genannte RND-Funktion ist in diesem Sinne eine reine Ausgabe-Funktion. Ausgegeben wird eine Zufallszahl zwischen 0 und 1. Die INT-Funktion hingegen ist eine Eingabe-Ausgabe-Funktion. Eingegeben wird eine Zahl, zum Beispiel 12.78, ausgegeben werden die Ziffern vor dem Komma, also 12 (Bild 2).

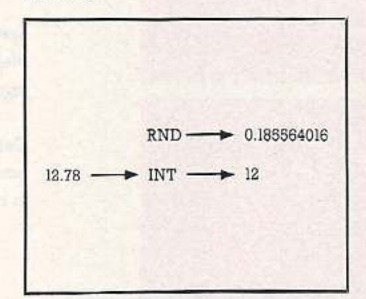


Bild 2. Zwei Grundtypen

Die Funktion INT erzeugt eine Zahl aus einer anderen Zahl; die Funktion LEFT$ erzeugt einen Text aus einem Text. Es geht aber auch »überkreuz«. Die Funktion LEN erhält als Eingabedatum einen Text und gibt eine Zahl aus, während umgekehrt die Funktion STR$ aus einer Zahl einen Text macht (Bild 3).

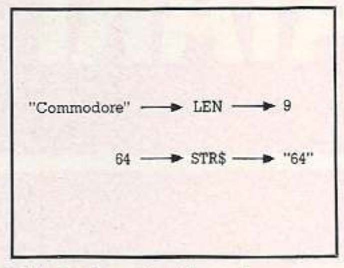


Bild 3. Eingabe: Text, Ausgabe: Zahl – und umgekehrt

Fassen wir zusammen: In (Commodore-) Basic finden wir die folgenden sechs Typen von Funktionen (Bild 4).

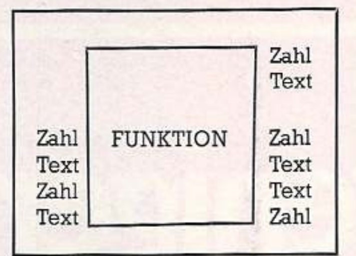


Bild 4. Basic-V.2.0 besitzt sechs Typen von Funktionen

Eine kleine Anmerkung noch: Die Befehle SPC und TAB, im Commodore-Handbuch unter der Überschrift »Funktionen« aufgeführt, sind in unserem Sinne keine Funktionen, da sie keine Daten erzeugen, sondern etwas bewirken, so wie zum Beispiel PRINT etwas bewirkt.

1.2 Mitteilungen an die Funktion

Wenn eine Funktion eine Zahl in eine andere umwandeln soll, dann muß ihr die Zahl in irgendeiner Weise mitgeteilt werden. Für Mitteilungen an Funktionen ist ein bestimmter Platz vorgesehen, nämlich die Klammern, die jedem Funktionsnamen (in Basic) folgen (Bild 5).



Bild 5. Einige Funktionen benötigen zusätzliche Angaben

Die Information, die eine Funktion braucht (sofern sie überhaupt eine braucht), kann von zweierlei Art sein: Es kann sich einmal um ein Eingabedatum handeln, das von der Funktion bearbeitet werden soll (zum Beispiel INT (12.78)), oder um Informationen darüber, wie die Funktion arbeiten soll. Daraus ergeben sich die folgenden Möglichkeiten (Bild 6).

ARTEN VON MITTEILUNG AN FUNKTIONEN
–
Eingabedatum
Arbeitshinweis
Eingabedatum und Arbeitshinweis

Bild 6. Arten von Mitteilung an Funktionen

Ein paar Beispiele zur Illustration:

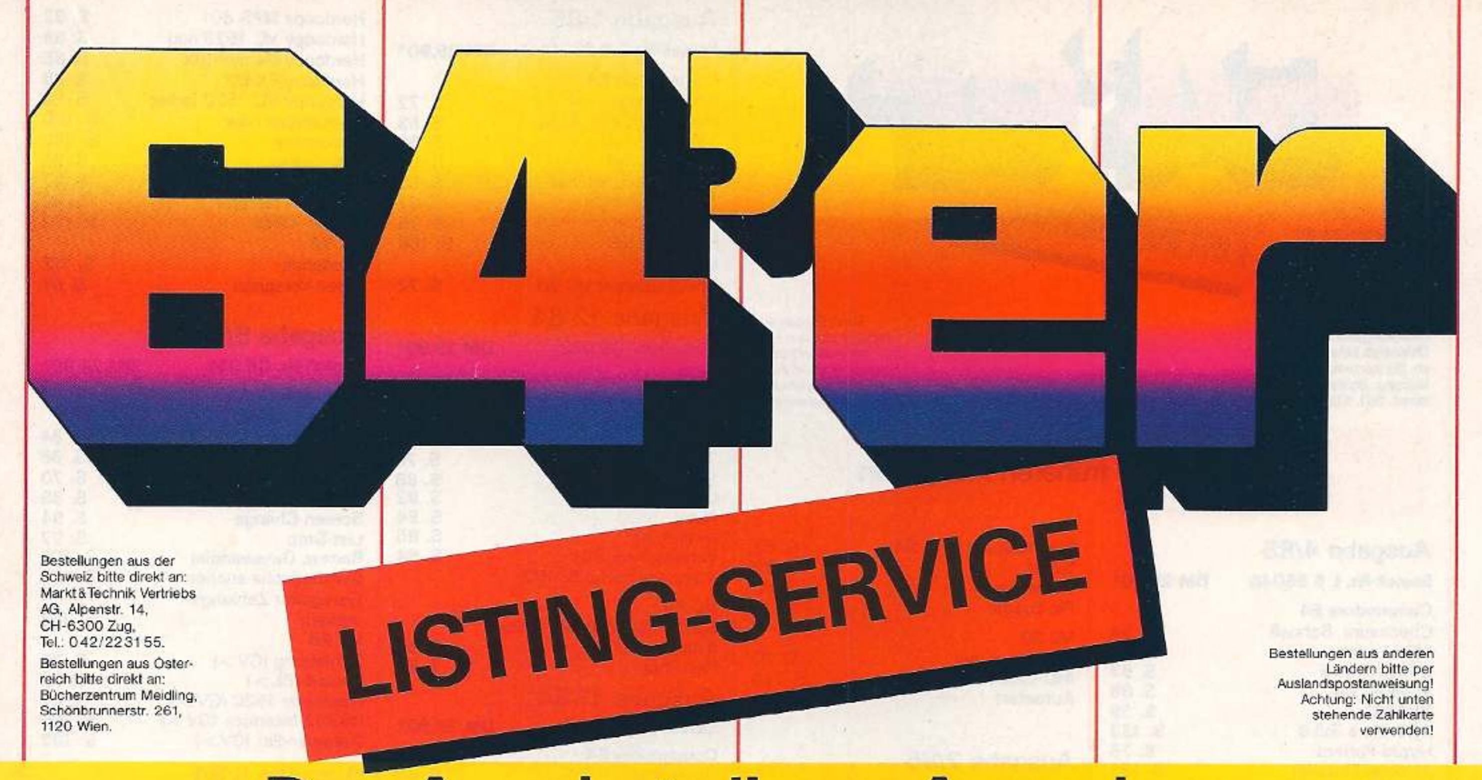
Die Funktion POS, die ausgibt, in welcher Bildschirmspalte sich der Cursor gerade befindet, weiß alles, was sie wissen muß, um ihre Aufgabe erfüllen zu können; der Programmierer muß ihr also keinerlei Informationen mitgeben. Da POS nun aber eine Funktion ist, ist ein Platz für Mitteilungen vorgesehen, das heißt der Programmierer muß die dem Funktionsnamen folgenden Klammern mit irgend etwas füllen. Der Einfachheit halber nimmt man dafür »0«: POS (0). Wenn Sie unbedingt wollen, können Sie auch irgend etwas anderes in die Klammern stecken, zum Beispiel »X« oder Ihren Namen – die Mitteilung landet in jedem Fall im Papierkorb, der Computer ignoriert sie.

Nicht im Papierkorb landet die Eingabezahl, die Sie zum Beispiel der Funktion SQR mitgeben. So berechnet SQR (25) die Wurzel aus 25, erzeugt also die Zahl 5.

Ebensowenig ignoriert wird eine Mitteilung, die Sie der Ausgabe-Funktion PEEK mitgeben. In diesem Fall wird die Information als Arbeitshinweis aufgefaßt: PEEK (2048) schaut in der Speicherzelle 2048 nach und sagt Ihnen dann, welchen Inhalt es gefunden hat.

Manchmal benötigt eine Funktion beide Arten von Informationen, so etwa die Funktion LEFT$. Sie muß zuerst wissen, was für ein String bearbeitet werden soll, und dann, wie lang der auszugebende String zu sein hat: LEFT$ ("Commodore",4) ergibt den Ausgabe-String "Comm". Mehrere Mitteilungen werden durch Kommata voneinander getrennt.

Natürlich können auch mehr als zwei Mitteilungen mitgegeben werden. Die Funktion MID$ benötigt, wie Sie wissen, im Normalfall drei: das Eingabedatum und zwei Arbeitshinweise (an welcher Stelle der Schnitt im Eingabe-String gemacht wer-



Bestellungen aus der Schweiz bitte direkt an: Markt & Technik Vertriebs AG, Alpenstr. 14, CH-6300 Zug, Tel.: 042/223155.

Bestellungen aus Österreich bitte direkt an: Bücherzentrum Meidling, Schönbrunnerstr. 261, 1120 Wien.

Bestellungen aus anderen Ländern bitte per Auslandspostanweisung! Achtung: Nicht unten stehende Zahlkarte verwenden!

Das Angebot dieser Ausgabe:

Die Diskette für eine Ausgabe kostet **29,90*** Mark. Sie werden bei einigen Disketten bestimmte Programme vermissen. Deren Autoren konnten sich nicht entschließen, ihr Programm im Rahmen des Leserservice für eine Verbreitung auf Datenträger freizugeben. Bei den Ausgaben 5 und 6 können noch Kassetten (VC ...) bestellt werden. Zu den Programmen sind immer die Seitenzahlen angegeben, unter der Sie die Beschreibungen in der entsprechenden Ausgabe finden können. Der Diskette liegen also keinerlei Informationen bei. Lesen Sie daher aufmerksam die Anleitung (ob SYS-Befehle nötig sind, in welcher Reihenfolge geladen werden muß, eventuelle Sprach- oder Speichererweiterungen und ähnliches mehr) in dem jeweiligen Artikel nach. Aus Aktualitätsgründen wird jeweils die abgedruckte Version angeboten. **Eventuelle systematische Fehler, die sich noch im Programm befinden können, müssen von Ihnen selbst, nach Studium des Druckfehlerteufelchens, korrigiert werden.**

Neu sind die Disketten-Symbole, die Sie am Anfang von verschiedenen veröffentlichten Listings finden können. Sie bedeuten, daß diese Programme auf der im Listing-Service angebotenen »Disk« enthalten sind. Der detaillierte Disk-Inhalt wird dann — wie gehabt — mit den Seitenzahlen in der nächsten Ausgabe abgedruckt.



Bestell-Nr. L 6 8505A

DM 29,90*

Bitte verwenden Sie für Ihre Bestellung **nur** die abgedruckte Postscheck-Zahlkarte zur Überweisung des Rechnungsbetrags. Sie erleichtern uns die Auftragsabwicklung und Sie **sparen** sich die **3,— DM** Versandkosten!

Einlieferungsschein/Lastschriftzettel

(nicht zu Mitteilungen an den Empfänger benutzen)

Gebühr für die Zahlkarte
(wird bei der Einlieferung bar erhoben)

bis 10 DM	— — —	90 Pf
über 10 DM (unbeschränkt)		1,50 DM

Bei Verwendung als Postüberweisung gebührenfrei

Bedienen Sie sich der Vorteile eines eigenen Postgirokontos

Auskunft hierüber erteilt jedes Postamt

Feld für postdienstliche Zwecke

Hinweis für Postgirokontoinhaber:
Dieses Formblatt können Sie auch als Postüberweisung benutzen, wenn Sie die stark umrandeten Felder zusätzlich ausfüllen. Die Wiederholung des Betrages in Buchstaben ist dann nicht erforderlich. Ihren Absender (mit Postleitzahl) brauchen Sie nur auf dem linken Abschnitt anzugeben.

1. Abkürzung für den Namen Ihres Postgiroamts (PGiroA) siehe unten
2. Im Feld »Postgiroteilnehmer« genügt Ihre Namensangabe
3. Die Unterschrift muß mit der beim Postgiroamt hinterlegten Unterschriftsprobe übereinstimmen
4. Bei Einsendung an das Postgiroamt bitte den Lastschriftzettel nach hinten umschlagen

Abkürzungen für die Ortsnamen der PGiroÄ:

Bln W	= Berlin West	Kln	= Köln
Dtmd	= Dortmund	Lshfn	= Ludwigshafen am Rhein
Esn	= Essen	Mchn	= München
Ffm	= Frankfurt am Main	Nbg	= Nürnberg
Hmb	= Hamburg	Sbr	= Saarbrücken
Han	= Hannover	Stgt	= Stuttgart
Klrh	= Karlsruhe		

Für Mitteilungen an den Empfänger

Bestellung Listing-Service **Wichtig:** Lieferanschrift (Rückseite) nicht vergessen!

Bestell-Nr.	Anzahl	x Einzelpreis	= Gesamtpreis
Summe bitte auf Vorderseite übertragen.		Gesamtsumme:	



Bestellungen aus der Schweiz bitte direkt an: Markt&Technik Vertriebs AG, Alpenstr. 14, CH-6300 Zug, Tel.: 042/223155.

Bestellungen aus Österreich bitte direkt an: Bücherzentrum Meidling, Schönbrunnerstr. 261, 1120 Wien.

Bestellungen aus anderen Ländern bitte per Auslandspostanweisung! Achtung: Nicht unten stehende Zahlkarte verwenden!

Programme aus früheren Ausgaben

Ausgabe 4/85

Bestell-Nr. L 6 8504A — DM 29,90*

Commodore 64	
Checksum. Schnell	S. 54
MSE Lader	S. 58
S-MON Teil 5	S. 67
Window 64	S. 88
XBASIC	S. 59
ASS. Kurs Teil 8	S. 138
Hypra-Perfect	S. 76
11 neue Einzeiler	S. 153
Print-List	S. 79
Netzgrafik	S. 71
ROT-DEMO	S. 73
X.Y-DEMO 1 u. 2	S. 73
Ostereier	S. 51
Befehlserweiterung	S. 23
VC 20	
Checksummer	S. 56
Supergrafik III	S. 73
Prg. Schutz	S. 83

Ausgabe 3/85

Bestell-Nr. L 6 8503A — DM 29,90*

Commodore 64	
Checksu. Schnell	S. 70
MSE Lader	S. 79
Hires-3/Teil 2	S. 136
Intellisearch$C	S. 153
Son of Destroyer	S. 42
Ligatab (LdM)	S. 63
Fl. Basicloader	S. 82
Floppylister	S. 82
E.Zeichnen 1 und 2	S. 72
Meisterschütze (AdM)	S. 53
Saver	S. 89
Screensave	S. 88
Pic-Loader	S. 90
VC 20	
Checksummer	S. 70
IRQ-Clock	S. 156
Autostart	S. 159

Ausgabe 2/85

Bestell-Nr. L 6 8502A — DM 29,90*

Commodore 64	
Checksummer 64	S. 65
MSE	S. 68
SMON (Teil 4)	S. 72
Grab des Pharao (LdM)	S. 56
Basic-Lader	S. 91
RAM-Floppy	S. 92
Notlandung	S. 156
Hires-3	S. 123
Benchmark-Test	S. 37
Format.Stringkur	S. 148
Versch. Test.USR	S. 140
Cursorsteuerung	S. 86
Rolling-Data	S. 88
Super-Memory	S. 81
VC 20	
Checksummer VC 20	S. 65
Familienplanung (AdM)	S. 53
Super 8-Steuerung	S. 70
Q+Bert (3K)	S. 78
Joypaint	S. 143
Bitmapping	S. 145
Multicolor-Demo	S. 146

Ausgabe 1/85

Bestell-Nr. L 6 8501A — DM 29,90*

Commodore 64	
Checksummer 64	S. 72
Handballtrainer (AdM)	S. 53
SMON (Teil 3)	S. 69
Hi-Eddi (LdM)	S. 57
Hypra-Load mal vier	S. 82
Tips und Tricks	S. 87
Provic 64	S. 76
Eingabe (UPB)	S. 156
VC 20	
Checksummer VC 20	S. 72

Ausgabe 12/84

Bestell-Nr. CB 022 — DM 29,90*

Commodore 64	
Synthesizer (AdM)	S. 51
SMON (2. Teil)	S. 60
3D-Vier gewinnt	S. 96
Trace	S. 76
Stringy	S. 88
Lader	S. 92
Auto	S. 84
Listschutz	S. 85
Simons Axo (SB)	S. 64
Kreuzworträtsel S. 150	
VC 20	
Mathematikal Basic (8K>) (LdM)	S. 55
Fast Tape	S. 80

Ausgabe 11/84

Bestell-Nr. CB 020 — DM 29,90*

Commodore 64	
Turtle Grafik (LdM)	S. 48
Schachmeister (AdM)	S. 50
SMON (1. Teil)	S. 59
Floppykurs	S. 117
FPLOT-Befehlserweiterung	S. 73
Get Koala pic	S. 66
Interrupttechnik	S. 84
Exsort (UPB)	S. 154
Einzeiler	S. 158
Simons Basic Befehlserweiterung (SB)	S. 90
VC 20	
Pseudosprites (8K)	S. 76
Laterna Magica (8K)	S. 68
Betriebssystem-Erweiterung (24K>)	S. 88
Supergrafik (GV)	S. 71
VC 20-Kurs (GV>)	S. 126

Ausgabe 10/84

Bestell-Nr. CB 019 — DM 29,90*

Commodore 64	
Finanzmathematik (AdM)	S. 68
Hypra-Load (LdM)	S. 67
Hardcopy Compact 2	S. 86
Hardcopy MPS 801	S. 82
Hardcopy VC 1526 neu	S. 83
Hardcopy Gemini-10X	S. 85
Hardcopy FX-80	S. 88
Hardcopy VC 1520 farbig	S. 84
Apocalypse now	S. 106
Supercopy	S. 102
Disk-Dump	S. 95
Diskettenorganisation	S. 97
User-Port-Tastatur	S. 92
Maske-(UPB)	S. 172
VC 20	
Epedemic	S. 112
Video-Vorspann	S. 81

Ausgabe 9/84

Bestell-Nr. CB 014 — DM 29,90*

Commodore 64	
Indexsequentielle Adreßdatei	S. 54
Spring Vogel (LdM)	S. 68
Orgel/Synthesizer (AdM)	S. 70
Sprite Aid +	S. 89
Screen Change	S. 94
List-Stop	S. 97
Renew, Datawandler	S. 102
Synthetische suchen	S. 104
Geregelter Zahlungsverkehr	S. 164
VC 20	
Schiebung (GV>)	S. 77
Deuzei (8K>)	S. 79
Hardcopy 1520 (GV>)	S. 87
RS232-Interface (GV>)	S. 100
Datawandler (GV>)	S. 102

Ausgabe 8/84

Bestell-Nr. CB 013 — DM 29,90*

Commodore 64	
Castle of Doom	S. 66
Pac-Boy	S. 89
Kopplung	S. 73
User-Port-Display	S. 97
RS232-Test	S. 77
View BAM	S. 99
Görlitz Hardcopy	S. 83
Milchvieh	S. 156
VC 20	
Kudiplo (3K)	S. 86
Print at Restore n (GV)	S. 101

Ausgabe 7/84

Bestell-Nr. CB 017 — DM 29,90*

Ausgabe 6/84

Commodore 64

Bestell-Nr. CB 018 — DM 29,90*

VC 20

Bestell-Nr. VC 008 — DM 29,90*

Bedeutung der Abkürzungen

- *LdM = Listing des Monats
- *AdM = Anwendung des Monats
- *SB = Simons Basic
- *GV = Grundversion
- *GV> = alle Speicherversionen können verwendet werden (einschließlich GV)
- *3K = 3-KByte-Speichererweiterung wird benötigt
- *8K> = Speichererweiterung größer als 8 KByte wird benötigt
- *UPB = Unterprogrammbibliothek

* Alle Preise inklusive Mehrwertsteuer.

Fehlende Hefte erhalten Sie bei: Markt & Technik Vertrieb 64'er Hans-Pinsel-Str. 2, 8013 Haar

---

Postgirokonto Nr. des Absenders

Empfängerabschnitt

DM | Pf

für Postgirokonto Nr.
14 199-803

Lieferanschrift und **Absender** der Zahlkarte

PLZ | Ort

Verwendungszweck

M&T Buchverlag Listing-Service

DM | Pf | für Postgirokonto Nr. **14 199-803**

Absender der Zahlkarte



PGiroA | Postgirokonto Nr. des Absenders | Postgiroteilnehmer

Zahlkarte/Postüberweisung
für maschinelle Beschriftung

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postgirokontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung siehe Rückseite)

DM | Pf

für **Markt&Technik** Verlag Aktiengesellschaft
in **8013 Haar**

Postgirokonto Nr.
14 199-803

Postgiroamt
München

Ausstellungsdatum | Unterschrift

Für Vermerke des Absenders

Postgirokonto Nr. des Absenders

Einlieferungsschein/Lastschriftzettel

DM | Pf

für Postgirokonto Nr. | Postgiroamt
14 199-803 | **München**

für **Markt&Technik** Verlag Aktiengesellschaft
Hans-Pinsel-Str. 2
in **8013 Haar**

den und wie lang der Ausgabe-String sein soll). MID$ ("Commodore",4,5) ergibt also "modor".

Vielleicht sollte zum Schluß noch angemerkt werden, daß Art und Anzahl der Informationen, die einer Funktion mitgeteilt werden können, natürlich nicht dem Belieben des Programmierers anheimgestellt, sondern für jede Funktion vorgegeben sind. Dasselbe gilt für die Reihenfolge, in der die verschiedenen Informationen angegeben werden.

Nun weiß eine Funktion also alles, was sie wissen muß, um ihre Arbeit zur Zufriedenheit des Programmierers zu tun — aber wohin mit dem Ergebnis, das sie erzeugt?

1.3 Wohin mit dem Ergebnis?

Das Datum, das eine Funktion erzeugt, eine Zahl oder ein Text, muß ihr irgendwie abgenommen werden.

Man kann das Ergebnis auf den Bildschirm bringen, zum Beispiel PRINT INT (12.78); man kann es einer Variablen zuordnen, zum Beispiel B$ = MID$ ("Commodore",4,5); man kann es für einen Vergleich benutzen, zum Beispiel IF PEEK (214) = 24 THEN PRINT CHR$ (147).

Mit einem Wort: Die durch Funktionen erzeugten Zahlen und Zeichenketten werden genauso verwendet wie Zahlen und Zeichenketten. Genauso wenig wie der Computer die isolierte Zahl 3.5 oder den isolierten String »Commodore« verstehen würde, genauso wenig versteht er ein alleinstehendes INT (12.78). Man muß ihm immer sagen, was er mit einer Zahl, einem Text oder einer Funktion (das heißt mit dem Ergebnis, das sie erzeugt) tun soll.

Und wozu überhaupt Funktionen?

1.4 Der Daseinszweck von Funktionen

Je mehr Funktionen eine Programmiersprache zur Verfügung stellt, um so leichter ist das Programmieren. Funktionen nehmen Programmierarbeit ab. Lassen Sie mich dies an zwei Beispielen illustrieren.

Beispiel 1: Die Funktion ABS

Die Funktion ABS erzeugt aus einer Zahl deren absoluten Wert, zum Beispiel aus 5 oder —5 den Wert 5. Ein Commodore-Programmierer schreibt also einfach:

110 WERT = ABS (ZAHL)

Es gibt aber Computer, die diese Funktion nicht kennen; in diesem Fall muß der Programmierer eine spezielle Programmroutine schreiben, und das sieht dann zum Beispiel so aus (Bild 7).

```
100 Y = ZAHL
110 GOSUB 40000 : REM ABSOLUT-
    WERT
120 WERT = Y
130 PRINT WERT
140 END
---------------------------------
40000 REM SUB: ABSOLUTWERT
40010      IF Y > = 0 THEN 40030
40020          Y = -Y
40030 REM ENDIF
40090 RETURN
```

Bild 7. Manche Funktionen müssen erst programmiert werden

Die armen Programmierer, die keinen Commodore haben! Andererseits sind wir Commodore-Programmierer arm dran, wenn wir zum Beispiel wissen wollen, ob die Zeichenkette B$ (»UTE«) in der Zeichenkette A$ (»COMPUTER«) enthalten ist, und wenn ja, ab welcher Stelle. Das geht ungefähr so (Bild 8).

```
100  TEXT$=A$
110  SUCH$=B$
120  LAENGE=LEN (TEXT$)—LEN
     (SUCH$) + 1
130  PO=0
140  GOSUB 40000:REM STRING
     SUCHEN
150...
--------------------------------
40000  REM SUB: STRING SUCHEN
40010    FOR I=1 TO LAENGE
40020      X$=MID$ (TEXT$,I,
           LEN (SUCH$))
40030      IF X$=SUCH$ THEN
           PO=I : I=LAENGE
40040    NEXT I
40090  RETURN
```

Bild 8. Die INSTR-Funktion

Beispiel 2: Die Funktion INSTR

PO enthält den Wert 5, was bedeutet, daß »UTE« gefunden wurde und in Position 5 beginnt. Wenn PO = 0 bleibt, dann wurde der Suchstring gefunden.

Ach, gäbe es doch eine Funktion, die uns diese ganze Programmierarbeit abnimmt! Tatsächlich gibt es sie bei anderen Computern, und sie heißt meist INSTR. Wenn der Programmierer sie hat, dann schreibt er zum Beispiel einfach

140 PO = INSTR (A$,B$)

Daß Funktionen das Programmieren erleichtern, daß sie dazu Programme kürzer, übersichtlicher und lesbarer machen, daß sie schließlich den Programmablauf beschleunigen, dürfte nun leicht einleuchten.

Es ist deshalb kein Wunder, daß es viele Versuche gibt, das eingebaute Basic durch zusätzliche Funktionen zu erweitern; zum Beispiel durch einzelne Routinen in Maschinensprache, die eine einzelne erwünschte Funktion zur Verfügung stellen; oder durch spezielle Basic-Erweiterungen wie Simons Basic (für den C 64) oder Exbasic Level II oder Macro Basic.

Wer Maschinensprache beherrscht und seinen Computer kennt, kann sich seine Erweiterungen jeweils nach Bedarf selber anfertigen. Wer beides nicht beherrscht, braucht aber auch nicht zu verzweifeln; denn, wie zu Beginn angedeutet: Es gibt auch in Basic die Möglichkeit, Funktionen selber zu basteln. Und wenn auch die Möglichkeiten von Commodore-Basic nicht das sind, was sie vielleicht sein könnten, sie sind noch immer größer als der Anfänger im allgemeinen weiß.

2. Selbstdefinierte Funktionen

2.1 Die möglichen Typen

Im Commodore-Basic können Funktionen, die Zahlen ausgeben, selber gemacht werden, und zwar sowohl vom Typ Ausgabe-Funktion als auch vom Typ Eingabe-Ausgabe-Funktion. Die letzteren sind auf die Eingabe von Zahlen beschränkt.

Die Mitteilungsmöglichkeiten sind ebenfalls beschränkt: Es kann maximal eine Information mitgegeben werden, und sie muß vom Typ »Eingabedatum« sein. Daß nur Zahlen Eingabedaten sein können, wurde schon erwähnt.

2.2 Die Syntax

Selbstgestrickte Funktionen werden gekennzeichnet durch FN, dem ein individueller Name folgt, zum Beispiel

FN KREISUMFANG

Für den Namen gelten die üblichen Regeln für Variablennamen, das heißt, nur die beiden ersten Zeichen eines Namens werden berücksichtigt. Die obige Funktion kann also ebensogut folgendermaßen geschrieben werden:

FN KR

Dem Funktionnamen FN KR folgen dann, wie bei Funktionen üblich, die Klammern, die für die Mitteilung eines eventuellen Eingabedatums zur Verfügung stehen.

2.3 Die Benutzung

Selbstgestrickte Funktionen werden genauso benutzt wie vorgefertigte. So könnte eine Programmroutine, die einen Würfel simuliert, so aussehen:

```
10 PRINT FN WUERFEL (0)
20 GOTO 10
```

Die Funktion FN WUERFEL ist eine reine Ausgabe-Funktion, weshalb ich als Mitteilung die nichtssagende 0 gewählt habe.

Die im folgenden besprochene Funktion FN KREISUMFANG ist eine Eingabe-Ausgabe-Funktion, der jeweils der Radius mitgeteilt werden muß. Ein Programm könnte so aussehen:

```
10 INPUT "WELCHES IST DER
RADIUS"; RD
20 UM = FN KREISUMFANG (RD)
30 PRINT "UMFANG BETRAEGT" UM
```

Aber woher weiß der Computer eigentlich, daß in der Funktion FN WUERFEL die Mitteilung in der Klammer ignoriert werden soll und daß sie andererseits bei der Funktion FN KREISUMFANG den Radius meint? Und woher weiß der Computer überhaupt, wie er die Ausgabe-Zahl erzeugen soll?

2.4 Die Definition

Bevor Sie eine selbstgestrickte Funktion einsetzen können, müssen Sie sie erst einmal definieren. Das muß logischerweise **vor** der Benutzung geschehen, am besten gleich zu Angang des Programms.

Dazu steht der Befehl DEF zur Verfügung. Lassen Sie uns zuerst die Funktion FN KREISUMFANG definieren. Das geht so:

DEF FN KREISUMFANG (RD) = 2 * PI * RD

Die Variable RD in der Klammer auf der linken Seite der »Gleichung« bezieht sich auf das Eingabedatum, also den Radius, der der Funktion mitgeteilt wird, wenn sie im Programm erscheint. Das Interessante dabei ist, daß der Radius nachher bei der Benutzung der Funktion keineswegs RD heißen muß. Man kann ihm jeden Namen geben, der einem in den Sinn kommt, und bei jedem Einsatz der Funktion kann man sich einen neuen einfallen lassen. Was allein wesentlich ist, das ist die Beziehung zwischen der Variablen RD in der Klammer auf der linken Seite und der Variablen RD auf der rechten Seite der Definitionsgleichung. Das bedeutet, daß man bei der Definition einer Funktion jede beliebige Variable benutzen kann. Es muß nur darauf geachtet werden, daß links und rechts dieselbe Variable benutzt wird. Die meisten Leute nehmen einfach X, was aber nicht in jedem Fall zu empfehlen ist. Ich komme darauf noch zurück.

Lassen Sie uns nun als nächstes die Ausgabe-Funktion FN WUERFEL definieren. Hier haben wir ein Problem: Wir geben ja dieser Funktion keine Information mit. Was also schreiben wir in die Klammer auf der linken Seite, die ja auf jeden Fall gefüllt werden muß? Nun, dieses Mal können wir ohne Bedenken X benutzen:

DEF FN WUERFEL (X) = INT (RND (1) * 6) + 1

Wie Sie sehen, erscheint auf der rechten Seite kein X. Aus dieser Tatsache schließt der Computer elektronenscharf, daß er bei der Benutzung dieser Funktion das, was in Klammern mitgeliefert wird, zu ignorieren hat.

In Commodore-Basic, so sahen wir, können wir einer selbstdefinierten Funktion also entweder gar keine Information mitgeben (Ausgabe-Funktion) oder ei-

nen Zahlenwert (Eingabe-Ausgabe-Funktion). Was aber, wenn wir zwei oder mehr Eingabedaten mitgeben möchten, sagen wir etwa bei einer Funktion FN RECHTECKINHALT? Nun, dies ist eben nicht möglich, aber wir können uns wie folgt aus der Affäre ziehen. Wir definieren zum Beispiel:

DEF FN RECHTECKINHALT (BREITE) = BREITE * LAENGE

Wenn wir die Funktion später aufrufen, müssen wir einfach dafür sorgen, daß die Länge dem Programm an dieser Stelle schon bekannt ist:

100 LAENGE = 5 : PRINT FN RECHTECKINHALT (BREITE)

Bei Funktionen dieser Art wird es vielleicht besonders deutlich: Wenn man eine Funktion benutzt, muß man sich darüber im klaren sein, welchen Eingabewert man ihr mitteilen muß. Deshalb ist es immer besser, bei der Definition einer Funktion »sprechende« Variablen zu benutzen statt des nichtssagenden X. Die folgende Definition verstehen Sie nach einem Jahr mit großer Wahrscheinlichkeit nicht mehr:

DEF FN A (X) = INT ((INT (X) + (X-INT (X)) * .6) + 100 + .5) / 100

Das bedeutungsleere X hat seine Berechtigung allein in reinen Ausgabe-Funktionen wie FN WUERFEL, und das ist auch die Konvention, an die ich mich selber halte.

Übrigens — wenn Sie einen Fehler bei der Definition einer Funktion machen, kann es sein, daß dieser erst beim Einsatz der Funktion angezeigt wird. Manch einen Anfänger hat dies schon zur Verzweiflung gebracht. Angenommen, Sie erhalten einen SYNTAX ERROR IN 220; Sie schauen sich die Zeile an:

220 PRINT FN A (5)

Sie können absolut keinen Fehler erkennen. Klar, der Fehler liegt ja auch ganz woanders, in Zeile 10 nämlich, wo Sie folgendermaßen definiert hatten:

10 DEF FN A (T) = T * WAND

Ihre Variable WAND enthält AND, was in Variablen nicht vorkommen darf, weil es ein Basic-Wort ist. Wenn also ein Syntax Error angezeigt wird für eine Zeile, die einen Funktionsaufruf enthält, schauen Sie sich zuerst einmal die dazugehörige Definition an, bevor Sie den Computer an die Wand werfen.

3. Der Wert ist der springende Punkt

Eigenbaufunktionen erzeugen Zahlen aus Zahlen, oder richtiger: sie erzeugen Zahlenwerte aus Zahlenwerten. Es ist wichtig, daß man sich folgendes ganz klar macht: Worauf es ankommt, ist der **Wert**. In welcher Form der Wert ausgedrückt wird, ist hingegen unerheblich. Das kann eine Zahl sein, aber ebenso eine Variable, ein mathematischer Ausdruck oder sogar eine Funktion. Die Länge des Radius eines Kreises könnte also zum Beispiel in einem Basic-Programm (und also auch im Zusammenhang mit Funktionen) folgendermaßen erscheinen:

10.5
RD
DURCHMESSER / 2
LEN (LINIE$)
FN HM (Y)

Bei der Definition einer Funktion ist also alles erlaubt — solange das Ergebnis ein Zahlenwert ist. Zwei Beispiele sollen dies deutlich machen.

Beispiel 1: Text zentrieren

Wir wollen eine Funktion definieren, die berechnet, ab welcher Bildschirmspalte ein Text gedruckt werden soll, um in der Mitte des Bildschirms zu erscheinen.

Beim C 64 hat die Bildschirmzeile 40 Spalten, die Mitte liegt bei Spalte 20. Die halbe Zeichenkette muß also vor der Mitte, die andere Hälfte nach der Mitte gedruckt werden. Die Funktion kann folgendermaßen definiert und benutzt werden:

10 DEF FN MITTE (X) = 20 — LEN (TEXT$) / 2
...
...
300 PRINT TAB(FN MITTE (0)); TEXT$

Unsere Funktion ist leider noch nicht vollkommen definiert, was deutlich wird, wenn die zu druckende Zeichenkette länger als die Bildschirmzeile ist, zum Beispiel 42 Zeichen lang. In diesem Fall erzeugt unsere Funktion ein negatives Ergebnis (—1), was TAB nicht verträgt, und was deshalb zu einer Fehlermeldung führt. Also müssen wir dafür sorgen, daß unsere Funktion nur dann rechnet, wenn die Zeichenkette gleich oder kleiner als 40 Zeichen lang ist. Wir könnten dieses Problem folgendermaßen lösen:

300 IF LEN (TEXT$) > 40 THEN PRINT TEXT$: GOTO 320
310 PRINT TAB(FN MITTE (0)); TEXT$
320 ...

Es gibt jedoch eine sinnvollere Möglichkeit, die es erlaubt, die Entscheidung, ob die Funktion rechnet oder nicht, sozusagen von der Funktion selber treffen zu lassen.

Dies erreichen wir, indem wir einen logischen Ausdruck in die Definition einbauen. Der Ausdruck

(LEN (TEXT$) <= 40)

ergibt den Wert —1, wenn er wahr ist, das heißt wenn TEXT$ 40 Zeichen lang ist oder kürzer. Wenn TEXT$ länger ist, ergibt der Ausdruck den Wert 0.

Unsere Definition lautet also:

10 DEF FN MITTE (X) = (20 — LEN (TEXT$) / 2) * ABS (LEN (TEXT$) <= 40)

FN MITTE ergibt den Wert 0, wenn TEXT$ länger als 40 Zeichen ist, und der Druck der Zeichenkette beginnt in der ersten Bildschirmspalte.

Das Beispiel zeigt, daß durchaus Strings in der Definition von Funktionen vorkommen können, wenn gewährleistet ist, daß das Endergebnis des Ausdrucks auf der rechten Seite der »Gleichung« ein Wert ist.

Beispiel 2: Kleinbuchstaben in Großbuchstaben verwandeln

Im Commodore-Basic kann die Definition einer Funktion höchstens eine Zeile lang sein. Diese Beschränkung läßt sich bis zu einem gewissen Grade umgehen, indem wir einfach mehrere Funktionen definieren und sie ineinander einbetten.

Es soll eine Funktion programmiert werden, mit deren Hilfe wir Kleinbuchstaben in Großbuchstaben verwandeln können. Wir benutzen dazu wieder logische Ausdrücke.

Wir gehen in zwei Schritten vor. Im ersten Schritt prüfen wir, ob das untersuchte Zeichen überhaupt ein Buchstabe ist. Das ist der Fall, wenn sein ASCII-Wert zwischen 65 und 93 oder 193 und 221 liegt:

10 DEF FN BU (Z) = (Z > 64 AND Z < 91) OR (Z > 192 AND Z < 219)

Das Ergebnis der Funktion FN BU ist —1, wenn der Ausdruck zutrifft. Das heißt, wenn die Funktion den Wert —1 erzeugt, dann ist das geprüfte Zeichen ein Buchstabe.

In Schritt 2 untersuchen wir, ob es sich um einen Kleinbuchstaben (dann muß er umgewandelt werden) oder einen Großbuchstaben handelt (dann darf er sich nicht verändern).

Kleinbuchstaben liegen zwischen 65 und 90. Das heißt, wenn der ASCII-Wert des geprüften Zeichens kleiner als 128 ist, haben wir es mit einem Kleinbuchstaben zu tun, und es muß 128 addiert werden. Das aber nur dann, wenn das Zeichen ein Buchstabe ist, das heißt, wenn FN BU den Wert —1 hat. Die Definition:

20 DEF FN KG (Z) = Z + (Z < 128) * 128 * FNBU (Z)

Die Funktion, die im Programm benutzt wird, ist natürlich lediglich die letztere; daß sie eine zweite Funktion enthält, braucht uns jetzt nicht mehr zu kümmern:

300 Z = ASC ("a")
310 Z = FN KG (Z)
320 PRINT CHR$ (Z)

Die Tatsache, daß es bei einer Funktion nur auf den Wert ankommt und nicht etwa darauf, daß dieser in Form einer Zahl ausgedrückt wird, betrifft nicht etwa nur die Definition von Funktionen, sondern auch ihre Anwendung. Was ich sagen will ist, daß es bei der einer Funktion mitgeteilten Information völlig gleichgültig ist, in welcher Form der mitgeteilte Wert ausgedrückt ist. Beispiele:

400 PRINT FN KG (65)
400 PRINT FN KG (A)
400 PRINT FN KG (A-128)
400 PRINT FN KG ("a")
400 PRINT FN KG (FN C (Y))

Ein Anwendungsbeispiel:

Um einen Namen, der in Kleinbuchstaben gespeichert ist, mit einem großen Anfangsbuchstaben zu versehen, könnte die Basic-Zeile 510 verwendet werden:

500 N$ = "commodore"
510 N$ = CHR$ (FN KG (ASC (N$))) + MID$ (N$,2)

ASC ergibt den ASCII-Wert des ersten Buchstabens von »commodore«, also 67; FN KG addiert 128 und erzeugt den Wert 195; CHR$ ergibt das Zeichen »C«; dies wird mit Hilfe von MID$ mit »ommodore« verknüpft und der Variablen N$ zugeordnet, die also schließlich die Zeichenkette »Commodore« enthält. Die ganze Transaktion wird übrigens nur mit Hilfe von (vorgefertigten und selbstdedinierten) Funktionen durchgeführt.

4. Legen Sie sich eine Funktionen-Bibliothek an

Die Möglichkeiten, die das Commodore-Basic für die Herstellung selbstdefinierter Funktionen zur Verfügung stellt, sind beschränkt — andere Basic-Dialekte sind da oft großzügiger. Da lassen sich Funktionen definieren, die auf Zeichenketten wirken; da können Definitionen viele Zeilen lang sein; da kann mehr als eine Information mitgegeben werden — aber wir wollen uns den Mund nicht wässrig machen.

Auch unsere Funktionen sind ein »mächtiges« Werkzeug (oder richtig deutsch) ein leistungsfähiges Werkzeug und keineswegs auf die Verarbeitung mathematischer Formeln beschränkt, wie man häufig annimmt. Man muß die Möglichkeiten nur nutzen.

Es lohnt sich, eine individuelle Bibliothek von Funktionsdefinitionen anzulegen, die man je nach Bedarf in seine Programme einbaut. Funktionen ersparen, wenn sie einmal zur Verfügung stehen, viel Programmierarbeit.

Im Anhang habe ich ein paar Funktionsdefinitionen zusammengestellt, die ich immer wieder in Programmen benutze. Da ist nichts Weltbewegendes dabei, aber ich habe mir dadurch schon manch unnötige, weil sich wiederholende, Denkarbeit erspart. Und das ist es, worauf es ankommt.

Welches sind **Ihre** Lieblingsfunktionen?

(Prof.Dr. Leuschner/gk)

Anhang: Einige einfache Funktionen zur Anregung

1. Zufallszahl zwischen 1 und ENDZAHL
DEF FN RD (ENDZAHL) = INT (RND (1) * ENDZAHL) + 1
PRINT FN RD (6) würfelt eine Zahl zwischen 1 und 6.
Anmerkung: Damit der Zufallsgenerator mit einer zufälligen Zufallszahl anfängt, sollte man zu Beginn des Programms folgende Zeile einfügen:
X = RND (—RND (0))

2. Zufallszahl zwischen ANFZAHL und ENDZAHL
DEF FN ZUFALL (ENDZAHL)
= INT(RND (1) * (ENDZAHL — ANFZAHL)) + ANFZAHL +1
ANFZAHL = 65 : PRINT CHR$ (FN ZUFALL (90)) erzeugt einen zufälligen Kleinbuchstaben.

3. Kommazahl zu Ganzzahl aufrunden
DEF FN AUFRUNDEN (ZAHL) = — INT (— ZAHL)
PRINT FN AUFRUNDEN (23.05) ergibt 24.
Anmerkung: Zum Abrunden benutzt man die einfache INT-Funktion.

4. Zahl mit festgelegter Anzahl von Nachkommastellen mit Rundung
DEF FN KOMMA (ZAHL)
= INT (ZAHL * 10 ↑ NACHKOMMA + .5) / 10 ↑ NACHKOMMA
NACHKOMMA = 2: PRINT FN KOMMA (25/6) ergibt 4.17.

5. Zahl mit festgelegter Anzahl signifikanter Ziffern (2 Funktionen)
DEF FN SG (ZAHL)
= 10 ↑ (1 — ZIFFERN + INT (LOG (ABS (ZAHL)) / LOG (10)))
DEF FN SIGNI (ZAHL)
= INT (ZAHL / FN SG (ZAHL) + .5) * FN SG (ZAHL)
ZIFFERN = 4 : PRINT FN SIGNI (ZAHL) ergibt bei ZAHL = 1234567 die Zahl 1235000, bei ZAHL = 12.345 die Zahl 12.35, etc.

6. Zahlen in einer Spalte drucken (PRINT US ING)
DEF FN US ING (ZAHL)
= SPALTE—ABS((ZAHL > 10)+(ZAHL > 100)+(ZAHL > 1000) + (ZAHL > 10↑4)+(ZAHL > 10↑5)+(ZAHL > 10↑6))
SPALTE = 20 : PRINT TAB(FN US ING (ZAHL)); ZAHL druckt Zahlen bis zu einer Million richtig als Kolonne.
Anmerkung: Wenn Sie die Variablen abkürzen, paßt die Definition in eine Zeile. Zwischen US und ING **muß** eine Leerstelle stehen!

7. Ungerade oder gerade Zahl?
DEF FN ODD (ZAHL) = ZAHL AND 1
PRINT FN ODD (25) ergibt den Wert 1, da 25 eine ungerade Zahl ist. Gerade Zahlen ergeben 0.

8. Modulus (Rest bei einer Division)
DEF FN MOD (ZAHL)
= INT (((ZAHL / TEILER — INT (ZAHL / TEILER)) * TEILER) + .5)
TEILER = 6: PRINT FN MOD (25) ergibt den Divisionsrest 1.

9. Uhrzeit dezimal darstellen
DEF FN DEZUHR (HR)
= INT ((INT (HR) + (HR — INT (HR)) / .6) * 100 + .5) / 100
H = FN DEZUHR (17.30) ergibt den Dezimalwert 17.5, mit dem man dann normal rechnen kann.

10. Dezimal ausgedrückte Uhrzeit als normale Uhrzeit darstellen
DEF FN UHR (DEZZT)
= INT ((INT (DEZZT) + (DEZZT — INT (DEZZT)) * .6) * 100 + .5) / 100
PRINT FN UHR (17.25) ergibt die normale Uhrzeit 17.15 Uhr.

11. ASCII-Code eines Zeichens in den Bildschirm-Code umwandeln
DEF FN SCREEN (AS) = (AS AND 128) / 2 OR (AS AND 63)
POKE 1024, FN SCREEN (ASC("A")) POKEt in die linke obere Ecke des C 64-Bildschirms den Buchstaben A.

12. Inhalt einer Bildschirmspeicherzelle lesen
DEF FN CRT (SPALTE)
= PEEK (1024 + (ZEILE — 1) * 40 + (SPALTE — 1))
ZEILE = 5 : PRINT FN CRT (20) ergibt den Inhalt der Speicherzelle der 20. Spalte in der 5. Zeile.

13. Exklusives Oder: Von zwei Bedingungen darf nur eine zutreffen.
Beispiel: Wenn Tante Amalie allein kommt, oder wenn Onkel Otto allein kommt, dann gehen wir auch zum Fest. Wenn aber beide kommen, dann gibt's Streit zwischen den beiden, also bleiben wir daheim. Wenn keiner von den beiden kommt, wird's langweilig, dann bleiben wir auch daheim. (Drei Funktionen werden dafür definiert.)
DEF FN B1 (X) = (A = AWERT) oder sonst eine Bedingung
DEF FN B2 (X) = (B = BWERT) oder sonst eine Bedingung
DEF FN EO R (X)
= ABS ((FN B1 (0) AND FN B2 (0)) < > (FN B1 (0) OR FN B2 (0)))
Als einander ausschließende Bedingungen seien A$ = "Amalie" und B$ = "Otto" für die beiden ersten Funktionen definiert worden. Dann ergibt die Funktion FN EO R den Wert 1 zum Beispiel bei folgender Situation:
A$ = "Amalie" : B$ = "Emilia" : PRINT FN EO R (0)

Anmerkung: Der Name der Funktion ist EO R, weil EOR das Basic-Wort OR enthielte, was zu einer Fehlermeldung führen würde.

14. ASCII-Wert eines Zeichens innerhalb eines Strings bestimmen
DEF FN AS C (PS) = ASC (MID$ (S$,PS,1))
S$ = "Commodore": PRINT FN AS C (6) ergibt den ASCII-Wert von »d«, also 68.

15. Einen Teilstring aus einem String »herausschneiden« und dessen Wert bestimmen
DEF FN WERT (PS) = VAL (MID$ (S$,PS,LAENGE))
S$ = "028255063": LAENGE = 3: PRINT FN WERT (4) ergibt den Wert 255.


64ER ONLINE

Wir suchen die Anwendung des Monats

Anwendung des Monats, was ist das? Nun, Sie haben einen Commodore 64 oder einen VC 20 und versuchen diesen irgendwie sinnvoll einzusetzen. Unter einer sinnvollen Anwendung versteht die 64'er Redaktion alles, was beispielsweise Programme im häuslichen Bereich bewirken. Es kann sich dabei um die Berechnung der Benzinkosten für Ihren Wagen handeln, um ein eigenes Textverarbeitungsprogramm gehen, sich um die Verwaltung Ihrer Tiefkühltruhe drehen oder ein ausgeklügeltes Telefon- und Adreßregister sein.

Setzen Sie Ihren VC 20/C 64 mehr oder weniger beruflich ein? Auch, oder vor allem, das ist eine sinnvolle Anwendung. Sie führen die Lohn- und Gehaltsabrechnung, Ihre Lagerverwaltung, die Bestellungen auf einem Commodore-Heimcomputer durch? So spezielle Anwendungen wie die Berechnung der Statik von selbstgezimmerten Regalen, von Klimadiagrammen oder Vokabellernprogrammen für den Schulunterricht oder die Zinsberechnung bei Krediten sind ebenfalls Themen, die mehr als konkurrenzfähig sind.

Uns ist die Anwendung des Monats

500 Mark

wert.

Schreiben Sie uns, was Sie mit Ihrem Computer machen:

Redaktion 64'er, Aktion: Anwendung des Monats, Hans-Pinsel-Str. 2, 8013 Haar bei München.

Einmal im Monat gibt es die SUPERCHANCE

Diese nicht einmalige Gelegenheit sollten Sie nutzen. Wie? Schicken Sie uns Ihr bestes, selbst erstelltes Programm. Bei der Art des Programms sind wir nicht wählerisch.

Sie haben ein sehr gutes (Schieß-, Knobel-, Denk-, Action-, Abenteuer-)Spiel geschrieben: einschicken!

Sie verfügen über ein komfortables Disketten-Kopier-(Sortier) Programm mit einigen außergewöhnlichen Leistungsmerkmalen: einschicken!

Sie haben das Basic um einige sinnvolle Befehle erweitert: einschicken!

Sie arbeiten mit einem selbsterstellten Textverarbeitungsprogramm, einer eigenen Tabellenkalkulation, einem semiprofessionellen Datenverwaltungsprogramm: einschicken!

Sie zeichnen und konstruieren mit einem selbsterstellten Programm in hochauflösender Grafik: einschicken!

Wir freuen uns über jeden Beitrag und honorieren mit bis zu

2000 Mark

für das Listing des Monats

Aus den besten Listings, die veröffentlicht werden, sucht die 64'er-Redaktion einmal im Monat das »Listing des Monats« aus. Alle Listings, die im 64'er abgedruckt sind, werden mit 100 bis 300 Mark honoriert. Die genaue Vorgehensweise beim Einsenden von Listings ist in dem Beitrag »Wie schicke ich meine Programme ein?« in verschiedenen Ausgaben beschrieben.

Schicken Sie Ihr Listing an: Redaktion 64'er, Superchance: Listing des Monats, Hans-Pinsel-Str. 2, 8013 Haar bei München.


64ER ONLINE

Fortsetzung von Seite 155
man nur 60 Abtastwerte pro Sekunde hat? Die Antwort gibt Bild 4. Man erhält den gestrichelten Verlauf mit einer Frequenz von nur 10 Hz. Noch seltsamer sieht das Resultat bei einer LFO-Frequenz von 25 Hz aus. Die Folge der Abtastwerte schwingt zwar ungefähr im Rhythmus von 25 Hz, dieser Bewegung ist aber zusätzlich ein Auf und Ab im 10-Hz-Rhythmus überlagert.

Der theoretische Hintergrund dieser Erscheinung sei hier nur gestreift: Nach dem Abtasttheorem muß die Abtastfrequenz mindestens doppelt so hoch sein, wie die höchste im abzutastenden Signal vorkommende Frequenz, damit die Abtastfolge dieses Signal richtig repräsentiert. Andernfalls weist die Abtastfolge Frequenzanteile auf, die im Originalsignal gar nicht vorkommen. Man nennt diesen Effekt Aliasing (von lat. alias = anderswo). In unserem Fall können die Bedingungen des Abtasttheorems nie vollständig erfüllt werden, da der ideale Sägezahn Obertöne beliebig hoher Ordnung enthält. Im ersten Fall von Bild 4 wird das Abtasttheorem grob verletzt: Die Abtastfrequenz ist bei weitem nicht doppelt so groß wie die Signalfrequenz. Als Resultat tritt nur eine Aliasing-Frequenz von 10 Hz auf. Im zweiten Fall wird das Abtasttheorem immerhin für die Grundschwingung des Signals erfüllt. 60 Hz ist mehr als doppelt so groß wie 25 Hz. Die 25 Hz sind in der Folge der Abtastwerte auch erkennbar. Die zweite Harmonische des Signals ist aber mit 50 Hz schon zu hoch für die Abtastung. Ihre Amplitude beträgt immerhin die Hälfte der Amplitude der Grundschwingung, wie eine Fourier-Analyse ergibt. Und genau diese Harmonische findet man auch hier als eine Aliasing-Frequenz von 10 Hz in der Folge der Abtastwerte wieder.

Aliasing tritt auch schon bei niedrigeren LFO-Frequenzen als 25 Hz auf. Der Effekt wird dann aber schwächer, weil die dafür verantwortlichen Obertöne von höherer Ordnung und damit von niedrigerer Amplitude sind. Mit dem Aliasing-Effekt kann man bei bewußtem Einsatz zusätzliche interessante Modulationen verwirklichen.

Wie es weitergeht

Nach diesem etwas anstrengenden theoretischen Teil werden wir uns in der nächsten Folge wieder der Tonerzeugung selbst zuwenden. Zunächst werden noch der Hüllkurvengenerator und der Portamento-Mechanismus von Modulator beschrieben, anschließend wird ein komfortables Editorprogramm vorgestellt, das ein schnelles, interaktives Manipulieren aller Modulator- und SID-Parameter ermöglicht. Mit dem Programm kann direkt über die Tastatur gespielt werden, und es können Sound-Parametersätze auf Diskette verwaltet werden. Dieses Programm soll dann in einer weiteren Folge zu einem kompletten dreistimmigen Sequenzer erweitert werden. Als Besonderheit wird dieses Programm unabhängige Melodier/Sound-Files erzeugen können, die für sich allein lauffähig sind. Die so erstellten Klangschöpfungen können dann in andere Programme eingebaut werden.

(Thomas Krätzig/aa)

VIC — Das »intelligente« Programm

Ein Computerprogramm zu schreiben, mit dem man sich einfach in normaler Umgangssprache unterhalten kann — das war die Aufgabe in unserem Programmierwettbewerb vom November '84. Ein Programm war »intelligener« als alle anderen.

Der Ausgangspunkt für diesen Programmierwettbewerb war die »Eliza-Story«. Im Jahre 1966 entwickelte Joseph Weizenbaum am Massachusetts Institute of Technology ein Program names »Eliza«, das — vereinfacht gesagt — einen Psychoanalytiker simuliert. Der Mensch begibt sich also in der Rolle des Patienten an die Computer-Tastatur und wird aufgefordert, von seinen Schwierigkeiten zu berichten. Aufgrund der Eingaben gibt Eliza dann durchaus differenzierte Antworten und stellt auch schon mal Zwischenfragen, so daß ein regelrechter Dialog zustande kommt. Das Eliza-Programm hat inzwischen eine große Verbreitung gefunden und existiert in unzähligen Versionen für alle gängigen Heimcomputer. Mit unserem Programmierwettbewerb wollten wir dazu anregen, ähnliche — und womöglich bessere — Programme für den C 64/VC 20 zu entwickeln. Wir erhielten auch eine ganze Reihe wirklich brauchbarer Programme — nur leider, leider handelte es sich bei vielen dieser Einsendungen um Programme, die eindeutig auf dem Original-Eliza basierten. Diese Programme gelangten natürlich gar nicht erst in die engere Wahl, denn bei unseren Wettbewerben ist immer noch die eigene Kreativität gefragt. Sieger wurde schließlich »VIC«, ein Programm, das sich in zwei wesentlichen Punkten von der Konkurrenz abhebt.

Zunächst einmal ist »VIC« sehr schnell. Auch bei längeren Eingaben werden für die Antwort selten mehr als vier bis fünf Sekunden gebraucht. »VIC« war damit, ob-

Der Autor von »VIC«, stellt sich vor

Mit meinen 37 Jahren zähle ich zwar nicht mehr zur jüngsten Hacker-Generation. Trotzdem bin ich ein begeisteter Computerspiele-Fan. Besonders gut gemachte Grafik-Adventures können mich stundenlang vor den Bildschirm fesseln. Mittlerweile ist auch meine Frau schon von der Adventuritis befallen, was schon mal dazu führen kann, daß das Abendessen erst nach Mitternacht stattfindet. Man muß doch vorher erst einmal aus diesem verflixten Tunnel herauskommen!

Das erste Mal kam mir vor 17 Jahren ein Computer in die Quere. Damals, nach abgeschlossener Berufslehre als Elektroniker, war ich im Studium etwas knapp bei Kasse und beschloß, diesem Mißstand mit einer Teilzeitarbeit zu begegnen. Es war in einem Platzreservations-System einer großen Fluggesellschaft. Zwei identische Computer-Anlagen waren dort installiert, um bei Ausfall des einen Computers sofort auf den anderen umschalten zu können. Dieses Umschalten war meine Aufgabe. Das kam dann so alle drei bis fünf Tage einmal vor. Die restliche Zeit konnte ich auf einem mitgebrachten Feldbett, neben dem Computer schlafen oder eben an meinem Studium weiterarbeiten. Dachte ich mir zumindest! Da stand aber die ganze Zeit einer der beiden Computer nutzlos herum und wartete nur darauf, von mir beschäftigt zu werden. Der Rest ist schnell erzählt: Einige Wochen später hatte ich mein Studium — für einen, für meine damaligen Verhältnisse, unwiderstehlichen Zahltag als Programmierer — an den Nagel gehängt.

Nach 10 Jahren EDV habe ich wieder zur Elektronik zurückgefunden. Seit 1978 besitze ich ein eigenes Geschäft und befasse mich mit der Entwicklung und dem Vertrieb von Medizin-Elektronik.

Meine Hobbies: Klavier (Jazz), Tennis, Ski, Schach, Computer-Spiele, Pokern.

(Robert Treichler)

wohl vollständig in Basic geschrieben, um ein vielfaches schneller als alle anderen Programme, die zum Teil sogar Maschinenroutinen verwendeten. Ein gutes Beispiel dafür, daß durch gut durchdachte Programmierung auch in Basic überraschend effektive Ergebnisse erzielt werden können. Zum anderen ist »VIC« sehr vielseitig. Man kann mit ihm über viele Themen reden, er bezieht sich in seinen Antworten in den meisten Fällen auf den Eingabesatz und manchmal sind seine Antworten nicht ohne Witz.

Natürlich ist das Programm — ebenso wie »Eliza« — nicht wirklich intelligent. Es sucht nach bestimmten Stichworten im Eingabesatz und erzeugt dann aus einer Reihe von Alternativen die Antworten, die mitunter gar nicht schlecht sind.

»VIC« gehört zu einer Minderheit der zu diesem Wettbewerb eingeschickten Programme — es handelt sich dabei nämlich um eines der wenigen VC 20-Listings, die uns zu diesem Wettbewerb erreichten. Es wurde geschrieben für den VC 20 mit mindestens 8 KByte Erweiterung, läuft aber ohne Änderungen auch auf dem C 64.

Hinweise zur Bedienung:

— Bei der Eingabe kann je nach Belieben über das Zeilenende hinausgeschrieben oder mit »—« getrennt werden.

— Es darf jeweils nur ein Satz eingegeben werden, der allerdings beliebige Satzzeichen enthalten kann.

— Ein Satz (und damit die Eingabe) wird abgeschlossen mit Return, Punkt, Ausrufezeichen oder Fragezeichen.

— Mit der DEL-Taste kann die Eingabe korrigiert werden. Alle anderen Steuerzeichen werden ignoriert. Tabelle 3 zeigt einen Beispieldialog mit »VIC«, wie sie ihn nach Eingabe des Listings ebenfalls führen können.

Ihr Computer als »intelligenter« Gesprächspartner — wenn das keine Abwechslung im Software-Eintopf ist?

(Robert Treichler/ev)

Hauptprogramm:		
10	— Initialisierung	
200	— Eingabe mit Suche nach Schlüsselworten	
300	— Bilden von Satz-Kompositionen + Texten	
400	— Verlegenheitsfragen stellen	
500	— Auf Bildschirm und Drucker ausgeben	
Routinen		
1000	— Eingabe-Unterprogramm	
2000	— Fragen u. Texte vorbereiten	
2500	— Für BS editieren	
3000	— DATAs einlesen/zählen	
10000	— DATAs	

Tabelle 1. »VIC«-Programmaufbau

AERGER
ARM
BIN
BIST
BLEIBE
BRAUCHE
BRUDER
BRUEDER
COMPUTER
DARF
FINDE
FRAU
FREUND
GEBE
GEHE
GELD
GESCHWISTER
GESUND
GLAUBE
GLUECK
HABE
HAETTE
HAST
HAT
HOFFE
IDIOT
KANN
KANNST
KOENNTE
KOMME
KRANK
KUMMER
LIEB
LUST
MACHE
MAENNER
MANN
MOECHTE
MOEGLICH
MUSS
MUTTER
ONKEL
REICH
SAG
SCHWAEGERIN
SCHWAGER
SCHWESTER
SEX
SOEHNE
SORGEN
SPIEL
SPINNER
STREIT
STRESS
TANTE
TOECHTER
TRAURIG
TROTTEL
UNGLUECK
UNZUFRIEDEN
VATER
VERWANDT
VIELLEICHT
WAERE
WEISS
WERDE
WETTER
WILL
WUERDE
WUNSCH

Tabelle 2. »VIC«-Schlüsselwörter

```
VIC - INTERVIEW

VIC:SALUE, ICH HEISSE VIC! WIE HEISST DU ?
DU: ICH HEISSE BOBO.
VIC:ALSO BOBO, ERZAEHLE MIR ETWAS VON DIR UND DEINER FAMILIE.
DU: TJA, ALSO ICH BIN 18 JAHRE ALT.
VIC:BIST DU ETWA STOLZ DARAUF, DASS DU 18 JAHRE ALT BIST?
DU: WARUM FRAGST DU DAS ?
VIC:DAS WEISST DU SICHER BESSER ALS ICH. ERZAEHLE WEITER!
DU: WARUM ERZAEHLST DU EIGENTLICH NICHTS VON DIR ?
VIC:KEINE AHNUNG! WAS ERWARTEST DU EIGENTLICH VON MIR?
DU: ICH ERWARTE EINEN ERHOEHTEN UNTERHALTUNGSWERT.
VIC:UEBRIGENS: WAS HAELST DU EIGENTLICH VON UNSERER KONVERSATION?
DU: SOLL ICH GANZ EHRLICH SEIN ?
VIC:DUMME FRAGE! WEISS DOCH JEDER!
DU: HEH, FREUNDCHEN, JETZT WIRST DU UNVERSCHAEHMT !
VIC:DU BIST ABER HOFFENTLICH NICHT VERHEIRATET - ODER?
DU: NEIN.
VIC:MACH GEFAELLIGST EINEN ANSTAENDIGEN SATZ!
DU: DAS BRAUCHE ICH MIR VON EINEM COMPUTER NICHT BIETEN ZU LASSEN !
VIC:WAS GLAUBST DU, SIND COMPUTER FUER DIE MENSCHHEIT EIN FLUCH ODER EIN SEGEN?
DU: ICH BIN MIR INZWISCHEN NICHT MEHR SICHER.
VIC:WEN INTERESSIERT DAS SCHON, DASS DU DIR INZWISCHEN NICHT MEHR SICHER BIST?
DU: DU BIST EIN SELTEN FRECHER KERL !
VIC:DUMMKOPF! DAS WEISS MAN DOCH, DASS ICH EIN SELTEN FRECHER KERL BIN.
DU: ICH WERDE DIR DEN STROM ABSCHALTEN !
VIC:WENN NUR ALLE MIR DEN STROM ABSCHALTEN WUERDEN!
DU: DU BIST WIRKLICH EIN TROTTEL !
VIC:NEBENBEI BEMERKT: SCHIMPFWOERTER MOECHTE ICH DANN KEINE MEHR HOEREN!
DU: OK, ICH WERDE DARAUF ACHTEN.
VIC:GLAUBST DU ES WAERE GUT, WENN DU DARAUF ACHTEN WUERDEST?
DU: WAS SOLL DIESE FRAGE ?
VIC:DEINE FRAGEREI GEHT MIR AUF DEN WECKER!
DU: DAS MASS IST BALD VOLL !
VIC:OK, ICH BIN WIRKLICH EIN TROTTEL  - WAS BIST DENN DU?
DU: WAHRSCHEINLICH AUCH EIN TROTTEL, DASS ICH MICH MIT DIR BESCHAEFTIGE.
VIC:VIELLEICHT BRAUCHST DU WIRKLICH DAS DIR VON EINEM COMPUTER NICHT BIETEN ZU L
ASSEN .
DU: ICH WARNE DICH !
VIC:GLAUBST DU AUCH, DASS ICH DIR ETWAS UEBERLEGEN BIN?
DU: AN GROESSENWAHN MANGELT ES DIR JEDENFALLS NICHT !
VIC:UNSERER KONVERSATION NACH ZU SCHLIESSEN, BIST DU NICHT GERADE DER HELLSTE!
DU: SEI NUR VORSICHTIG !
VIC:SO, NUN MUSS ICH ABER SCHLUSS MACHEN. TSCHUESS BOBO
```

Tabelle 3. Ein typischer Dialog mit »VIC«

```
1 REM 'VIC' EIN KI-PROGRAMM F. VC20 + 8KB   <033>
2 REM       ROBERT TREICHLER                <001>
3 REM FL-9497 TRIESENBERG (F.TUM LIECHTENS
  TEIN)                                     <242>
10 REM                                      <153>
20 REM INIT.+TITEL                          <178>
30 REM                                      <173>
50 PRINT"{CLR,7SPACE}V I C{3DOWN}"          <183>
60 PRINT"KOPIE AUF DRUCKER?{4SPACE,DOWN}<J
   /N>"                                     <014>
70 GET Z$:IF Z$=""THEN 70                   <172>
80 IF Z$="N"THEN 100                        <213>
90 D=1:OPEN 4,4:PRINT#4,CHR$(14)"VIC - INT
   ERVIEW"CHR$(15)+CHR$(10)                 <078>
100 PRINT"{CLR,3DOWN}ICH MUSS MICH MAL KUR
    Z{DOWN}KONZENTRIEREN.":EA=50:FA=50:SV=
    3:GOSUB 3200                            <154>
110 DIM A%(25),E$(EA),F$(FA),S$(SA),S%(SA,
    SV),T$(TK+TN),T2$(TK),TV$(TV),TF$(TF)   <066>
120 GOSUB 3000:F$="SALUE, ICH HEISSE VIC!
    WIE HEISST DU ?":GOSUB 2500:GOSUB 1000  <067>
130 N$=E$(W):F$="ALSO "+N$+", ERZAEHLE MIR
     ETWAS VON DIR UND DEINER FAMILIE.":GO
    TO 500                                  <095>
197 REM                                     <084>
198 REM INPUT+VERGL.                        <190>
199 REM                                     <086>
200 GOSUB 1000:FOR I=0 TO W:A=ASC(E$(I))-6
    5:IF A<0 OR A>25 THEN 290               <136>
210 S=A%(A):IF S=0 THEN 290                 <011>
220 FOR S=S TO SA:IF S%(S,0)>TK THEN Z$=LE
    FT$(E$(I),LEN(S$(S))):GOTO 250          <034>
230 Z$=E$(I)                                <025>
250 IF Z$>S$(S)THEN NEXT S:GOTO 290         <169>
260 IF Z$<S$(S)THEN 290                     <030>
270 GOSUB 2000                              <094>
290 NEXT I                                  <238>
297 REM                                     <185>
298 REM OUTPUT VORBER.                      <169>
299 REM                                     <187>
310 IF FF=0 THEN 350                        <111>
320 FF=0:IF TF>0 THEN TF=TF-1               <088>
330 F$=TF$(TF):GOTO 500                     <104>
350 IF W<2 AND F<1 THEN F$="MACH GEFAELLIG
    ST EINEN ANSTAENDIGEN SATZ!":GOTO 500   <225>
360 IF F<1 THEN 400                         <089>
370 F$=F$(F):F$(F)="":F=F-1:GOTO 500        <109>
397 REM                                     <029>
398 REM VERLEGENHEITSFRAGEN STELLEN         <189>
399 REM                                     <031>
400 IF TA<3 THEN 430                        <213>
410 F$="SO, NUN MUSS ICH ABER SCHLUSS MACH
    EN. TSCHUESS "+N$                       <125>
420 GOSUB 2500:CLOSE 4:END                  <193>
430 T=INT(RND(1)*TV):IF TV$(T)=""THEN 430   <186>
460 TA=TA+1:F$=TV$(T):TV$(T)=""             <240>
497 REM                                     <129>
498 REM FRAGE AUSGEBEN                      <049>
499 REM                                     <131>
500 GOSUB 2500                              <073>
510 GOTO 200                                <026>
990 REM                                     <112>
991 REM***************                      <231>
992 REM                                     <114>
993 REM ROUTINEN:                           <033>
994 REM                                     <116>
995 REM                                     <117>
997 REM INPUT                               <007>
998 REM                                     <120>
999 REM GET CHAR                            <119>
1000 E=0:E$="":GET A$:IF A$>""THEN 1000     <114>
1010 PRINT"{RVSON,SPACE,RVOFF,LEFT}";       <078>
1020 GET A$:IF A$=""THEN 1020               <143>
1030 A=ASC(A$):IF A<>20 THEN 1100           <174>
1040 IF E<1 THEN 1020                       <050>
1050 E=E-1:E$=LEFT$(E$,E):PRINT"{SPACE,2LE
     FT}";:GOTO 1010                        <143>
1100 IF A=13 THEN A$="."                    <098>
1110 IF A$<" "OR A$>"Z"THEN 1020            <015>
1120 E$=E$+A$:E=E+1:PRINT A$;               <187>
1130 IF A$="!"OR A$="."THEN 1200            <024>
1140 IF A$="?"THEN FF=1:GOTO 1200           <057>
1150 GOTO 1010                              <205>
1199 REM IN WORTE ZERLEGEN                  <198>
1200 IF E<1 THEN RETURN                     <157>
```

Listing »VIC« — Das »intelligente« Programm

```
1202 PRINT:PRINT"{2DOWN}LASS MICH UEBERLEG
     EN."                                   <232>
1205 W=-1:WA=0:FOR I=1 TO E:A$=MID$(E$,I,1
     ):A=ASC(A$)                            <114>
1210 IF A>64 AND A<91 OR A>47 AND A<58 THE
     N 1300                                 <058>
1220 IF WA=0 OR WT=1 THEN 1350              <123>
1230 IF A=45 THEN WT=1:GOTO 1350            <122>
1250 WA=0:GOTO 1350                         <236>
1300 IF WA=0 THEN WA=1:W=W+1:E$(W)="":IF W
     >EA-1 THEN RETURN                      <090>
1310 WT=0:E$(W)=E$(W)+A$                    <205>
1350 NEXT                                   <205>
1399 REM 1.PERS ERSETZEN DURCH 2.PERS.      <082>
1400 FOR I=0 TO W:IF LEN(E$(I))>6 THEN 149
     0                                      <092>
1405 Z$=LEFT$(E$(I),4)                      <046>
1410 IF Z$="ICH"THEN E$(I)="DU":GOTO 1450   <032>
1415 IF Z$="DU"THEN E$(I)="ICH":GOTO 1450   <037>
1420 IF Z$="MICH"THEN 1480                  <037>
1425 IF Z$="DICH"THEN 1470                  <032>
1430 IF Z$="MEIN"THEN 1480                  <055>
1435 IF Z$="DEIN"THEN 1470                  <050>
1440 IF Z$="MIR"THEN 1480                   <000>
1445 IF Z$="DIR"THEN 1470                   <251>
1450 GOTO 1490                              <006>
1470 E$(I)="M"+RIGHT$(E$(I),LEN(E$(I))-1):
     GOTO 1490                              <128>
1480 E$(I)="D"+RIGHT$(E$(I),LEN(E$(I))-1)   <240>
1490 NEXT                                   <089>
1500 REM DRUCKEN                            <124>
1520 IF D THEN PRINT#4,"DU: ";E$            <030>
1530 RETURN                                 <141>
1997 REM                                    <099>
1998 REM FRAGEN VORBER.                     <021>
1999 REM                                    <101>
2000 IF F>FA-1 THEN RETURN                  <241>
2010 REM NICHT BENUTZTE TEXTE SUCHEN        <167>
2020 Z=0:FOR J=0 TO SV:T=S%(S,J):IF T$(T)>
     ""THEN Z(Z)=T:Z=Z+1                    <049>
2030 NEXT J                                 <193>
2050 IF Z=0 THEN RETURN                     <006>
2060 T=Z(RND(1)*Z):IF T=<TK AND I=W THEN R
     ETURN                                  <091>
2100 REM TEXT HOLEN                         <134>
2120 F=F+1:F$(F)=T$(T):T$(T)="":IF T>TK TH
     EN RETURN                              <094>
2200 REM TEXT-KOMPOSITION                   <253>
2210 IF E$(I+1)="ICH"OR E$(I+1)="DU"THEN 2
     230                                    <104>
2220 FOR J=I+1 TO W:GOTO 2250               <220>
2230 FOR J=0 TO W:IF J=I THEN J=J+2:REM IN
     VERSION                                <191>
2250 IF E$(J)="UND"OR E$(J)="ODER"THEN J=W
     :GOTO 2270                             <154>
2260 F$(F)=F$(F)+" "+E$(J)                  <044>
2270 NEXT J                                 <178>
2280 F$(F)=F$(F)+" "+T2$(T)                 <139>
2290 RETURN                                 <136>
2497 REM                                    <089>
2498 REM BS EDITIEREN                       <136>
2499 REM                                    <091>
2500 PRINT"{CLR}":X$=F$                     <015>
2510 Z=22                                   <071>
2520 Z$=MID$(F$,Z,1):IF Z$=""THEN 2550      <037>
2530 IF ASC(Z$)<65 THEN 2550                <156>
2540 Z=Z-1:GOTO 2520                        <195>
2550 Z$=LEFT$(F$,Z):PRINT Z$:IF Z<22 THEN
     PRINT"{DOWN}";                         <143>
2560 IF Z=>LEN(F$)THEN 2580                 <070>
2570 F$=RIGHT$(F$,LEN(F$)-Z):IF F$<>""THEN
      2510                                  <169>
2580 PRINT"{2DOWN}":IF D=1 THEN PRINT#4,"V
     IC: ";X$                               <191>
2590 RETURN                                 <182>
2997 REM                                    <079>
2998 REM DATA'S LESEN                       <091>
2999 REM                                    <081>
3000 RESTORE:FOR I=1 TO SA:READ S$(I):J=0:
     A=ASC(S$(I))-65:IF A%(A)=0 THEN A%(A)
     =I                                     <074>
3030 READ Z:IF Z=0 THEN 3050                <050>
3040 S%(I,J)=ABS(Z)-TK*(Z>0):J=J+1:GOTO 30
     30                                     <236>
3050 NEXT:READ Z$                           <182>
3060 FOR I=TK+1 TO TK+TN:READ T$(I):NEXT:R
```

```
      EAD Z$                                 <082>
3070 FOR I=1 TO TK:READ T$(I),T2$(I):NEXT:
      READ Z$                                <055>
3080 FOR I=0 TO TV-1:READ TV$(I):NEXT:READ
      Z$                                     <014>
3090 FOR I=0 TO TF-1:READ TF$(I):NEXT:READ
      Z$:IF Z$<>"$"THEN PRINT"{CLR}DATA-FE
      HLER{DOWN}":END                        <230>
3100 RETURN                                  <182>
3200 REM                                     <027>
3201 REM DATA'S ZAEHLEN                      <183>
3202 REM                                     <029>
3210 GOSUB 3300:SA=Z:GOSUB 3300:TN=Z:GOSUB
      3300:TK=Z/2:GOSUB 3300:TV=Z:GOSUB 33
      00:TF=Z:RETURN                         <059>
3300 Z=0                                     <044>
3320 READ Z$:IF Z$="$"THEN RETURN            <155>
3330 A=ASC(Z$):IF A>57 OR A<45 OR A=46 THE
      N Z=Z+1                                <188>
3340 GOTO 3320                               <106>
9999 REM *************                       <231>
10000 REM                                    <198>
10010 REM SCHLUESSELWOERTER + VERKETTUNG     <045>
10020 REM                                    <218>
10650 DATA AERGER,14,25,0,ARM,2,22,0         <210>
10660 DATA BIN,-1,-12,-13,-19,0,BIST,-17,-
      35,-37,-38,0,BLEIBE,-7,-30,0           <133>
10662 DATA BRAUCHE,-14,0,BRUDER,9,0,BRUEDE
      R,9,0                                  <013>
10670 DATA COMPUTER,11,0                     <179>
10680 DATA DARF,-3,-29,0                     <045>
10700 DATA FINDE,-34,0,FRAU,3,4,0,FREUND,1
      5,0                                    <077>
10710 DATA GEBE,-32,0,GEHE,-15,0,GELD,2,0,
      GESCHWISTER,9,0                        <070>
10712 DATA GESUND,21,22,0,GLAUBE,-5,27,29,
      0,GLUECK,23,0                          <139>
10720 DATA HABE,-2,-22,-39,0,HAETTE,17,0,H
      AST,-36,0,HAT,-25,0,HOFFE,-6,0         <250>
10730 DATA IDIOT,20,0                        <249>
10750 DATA KANN,-4,-24,0,KANNST,-18,-42,0,
      KOENNTE,17,0                           <130>
10752 DATA KOMME,-33,0,KRANK,22,0,KUMMER,2
      5,0                                    <188>
10760 DATA LIEB,6,28,0,LUST,24,0             <135>
10770 DATA MACHE,-11,-21,-27,-28,0,MAENNER
      ,5,0,MANN,5,0                          <119>
10772 DATA MOECHTE,-16,-31,0,MOEGLICH,13,0
      ,MUSS,-20,-26,0,MUTTER,7,0             <023>
10790 DATA ONKEL,10,0                        <053>
10820 DATA REICH,2,22,0                      <166>
10830 DATA SAG,12,29,0,SCHWAEGERIN,10,0,SC
      HWAGER,10,0                            <006>
10832 DATA SCHWESTER,9,0,SEX,26,0,SOEHNE,1
      ,0,SORGEN,25,0,SPIEL,16,0              <170>
10834 DATA SPINNER,20,0,STREIT,25,0,STRESS
      ,25,0                                  <253>
10840 DATA TANTE,10,0,TOECHTER,1,0,TRAURIG
      ,14,25,0,TROTTEL,20,0                  <187>
10850 DATA UNGLUECK,14,23,0,UNZUFRIEDEN,14
      ,0                                     <083>
10860 DATA VATER,8,0,VERWANDT,10,0,VIELLEI
      CHT,13,0                               <220>
10870 DATA WAERE,17,0,WEISS,-10,0,WERDE,-9
      ,-23,-40,-41,0,WETTER,18,0             <022>
10872 DATA WILL,-8,-9,-41,0,WUERDE,17,0,WU
      NSCH,19,0                              <120>
19990 DATA$                                  <011>
19997 REM                                    <250>
19998 REM TEXTE                              <133>
19999 REM                                    <252>
20001 DATA DU BIST SICHER STOLZ AUF DEINE
      KINDER. WAS MACHEN SIE?                <250>
20002 DATA GELD ALLEIN MACHT NICHT GLUECKL
      ICH!                                   <163>
20003 DATA ICH GLAUBE FRAUEN SIND EIN HEIK
      LES THEMA.                             <229>
20004 DATA"UEBRIGENS, WIE SOLLTE DEINE TRA
      UMFRAU SEIN?"                          <075>
20005 DATA MACHST DU DIR VIEL AUS MAENNER? <140>
20006 DATA MIT DER LIEBE IST ES HALT SO EI
      NE SACHE.                              <035>
20007 DATA ERZAEHLE MIR MEHR UEBER DEINE M
      UTTER.                                 <067>
20008 DATA WAR DEIN VATER SEHR STRENG MIT
      DIR?                                   <146>
20009 DATA ERZAEHL MIR ETWAS MEHR VON DEIN
      EN GESCHWISTERN.                       <007>
20010 DATA GIBT ES IN DEINER VERWANDTSCHAF
      T AUCH LEUTE DIE DU MAGST?             <241>
20011 DATA"WAS GLAUBST DU, SIND COMPUTER F
      UER DIE MENSCHHEIT EIN FLUCH ODER EI
      N SEGEN?"                              <144>
20012 DATA DAS HALTE ICH FUER EIN GERUECHT
      .                                      <170>
20013 DATA DU SCHEINST ETWAS UNSICHER ZU S
      EIN.                                   <233>
20014 DATA VERSUCHE SOLCHE NEGATIVEN GEDAN
      KEN VON DIR FERN ZU HALTEN.            <150>
20015 DATA DU BIST ABER HOFFENTLICH NICHT
      VERHEIRATET - ODER?                    <094>
20016 DATA GAMBLER!                          <028>
20017 DATA SO SICHER SCHEINT DAS ABER NICH
      T ZU SEIN - ODER?                      <076>
20018 DATA HAST DU KEIN BESSERES THEMA ALS
       DAS WETTER?                           <144>
20019 DATA WUENSCHE SIND DIE TRIEBFEDER DE
      R MENSCHHEIT. (GUT - WAS?)             <203>
20020 DATA"NEBENBEI BEMERKT: SCHIMPFWOERTE
      R MOECHTE ICH DANN KEINE MEHR HOEREN
      !"                                     <107>
20021 DATA"A VOTRE SANTE! (HAST DU GESEHEN
      , ICH KANN SOGAR FRANZOESISCH.)"       <049>
20022 DATA"DU SAGST DIR VERMUTLICH AUCH: L
      IEBER GESUND UND REICH, ALS KRANK UN
      D ARM."                                <230>
20023 DATA"DU KENNST DOCH DIE GESCHICHTE V
      OM HANS IM GLUECK, ODER?"              <129>
20024 DATA"APROPOS LUST: ICH HAETTE JETZT
      GERADE LUST AUF EIN BIER."             <011>
20025 DATA DU SOLLTEST DAS LEBEN ETWAS VON
       DER HEITEREN SEITE NEHMEN.            <100>
20026 DATA WEISST DU WIE NONNEN ZAEHLEN?
                1 2 3 4 5 PFUI!              <206>
20027 DATA DAS KANN ICH ALLERDINGS NICHT S
      O RECHT GLAUBEN.                       <127>
20028 DATA GEHT DIE LIEBE BEI DIR AUCH DUR
      CH DEN MAGEN?                          <044>
20029 DATA GLAUBST DU DAS WIRKLICH?          <045>
29990 DATA$                                  <066>
29997 REM                                    <049>
29998 REM KOMPOSITIONS-TEXTE                 <152>
29999 REM                                    <051>
30001 DATA"BIST DU ETWA STOLZ DARAUF, DASS
       DU",BIST?                             <069>
30002 DATA"WAS FUER EIN ZUFALL, AUCH ICH H
      ABE",.                                 <192>
30003 DATA"WER BESTIMMT DENN, DASS DU",DAR
      FST?                                   <006>
30004 DATA"DAS HABE ICH MIR GEDACHT,DASS D
      U",KANNST.                             <029>
30005 DATA WARUM GLAUBST DU,?                <207>
30006 DATA"HOFFST DU NOCH ETWAS ANDERES, A
      USSER",?                               <237>
30007 DATA"DEINE STANDHAFTIGKEIT IN EHREN.
       IST ES ABER WIRKLICH KLUG,",ZU BLEI
      BEN?                                   <210>
30008 DATA"AN WAS DENKST DU, WENN DU",WILL
      ST?                                    <151>
30009 DATA"GLAUBST DU ES WAERE GUT, WENN D
      U",WUERDEST?                           <247>
30010 DATA"WIE MEINST DU DAS GENAU, DU WUE
      SSTEST",?                              <019>
30011 DATA ALSO MACH,. IST MIR AUCH GLEICH
      .                                      <168>
30012 DATA"WEN INTERESSIERT DAS SCHON, DAS
      S DU",BIST?                            <227>
30013 DATA"OH, ICH WAERE AUCH GERNE",.       <112>
30014 DATA VIELLEICHT BRAUCHST DU WIRKLICH
      ,.                                     <203>
30015 DATA ICH MOECHTE EIGENTLICH AUCH,GEH
      EN.                                    <206>
30016 DATA"AN WAS DENKST DU, WENN DU",MOEC
      HTEST?                                 <108>
30017 DATA"UEBRIGENS:DU BIST SELBST",!       <064>
30018 DATA"WETTEN WIR, DASS DU SELBST",KAN
      NST?                                   <062>
30019 DATA"SO, SO. DU BIST ALSO",. OB DAS
      WOHL JEMANDEN JUCKT?                   <022>
30020 DATA"KANN ICH DIR HELFEN, WENN DU DA
```

Listing »VIC« (Fortsetzung)


64ER ONLINE

```
      S NAECHSTE MAL",MUSST?                <014>
30021 DATA"ERZAEHLE MIR MEHR DARUEBER, WIE
      DU",MACHST.                           <243>
30022 DATA"WAS GLAUBST DU WOHER DAS KOMMT,
       DASS DU",HAST?                       <136>
30023 DATA WARUM WIRST DU,?                 <104>
30024 DATA"KANNST DU MIR ERKLAEREN, WIE MA
      N",KANN?                              <237>
30025 DATA WER HAT SONST NOCH,?             <054>
30026 DATA DU BIST NICHT DER EINZIGE. AUCH
       ICH SOLLTE,.                         <185>
30027 DATA"WAS DENKST DU DIR DABEI, WENN D
      U",MACHST?                            <048>
30028 DATA"WAS WAERE WOHL, WENN JEDER",MAC
      HEN WUERDE?                           <210>
30029 DATA"DAS MUESSTE MIR MAL EINER SAGEN
      , DASS ICH",DARF!                     <191>
30030 DATA WARUM KANNST DU NUR SO STUR SEI
      N UND,BLEIBEN?                        <145>
30031 DATA" "," - DAS MOECHTE NOCH MANCHER
      !"                                    <076>
30032 DATA"ICH FINDE ES GROSSZUEGIG VON DI
      R, DASS DU",GIBST.                    <076>
30033 DATA KOMMT SONST NOCH JEMAND,?        <170>
30034 DATA ICH STIMME DIR VOLL BEI. AUCH I
      CH FINDE,.                            <188>
30035 DATA"OK, ICH BIN"," - WAS BIST DENN
      DU?"                                  <185>
30036 DATA"DAS WEISS DOCH JEDER, DASS ICH"
      ,HABE.                                <016>
30037 DATA"HAT LANGE GEDAUERT, BIS DU GEME
      RKT HAST, DASS ICH",BIN.              <102>
30038 DATA"DUMMKOPF! DAS WEISS MAN DOCH, D
      ASS ICH",BIN.                         <209>
30039 DATA"BIST DU FROH, DASS DU",HAST?     <024>
30040 DATA WER WIRD SONST NOCH,?            <158>
30041 DATA WENN NUR ALLE,WUERDEN!           <003>
30042 DATA WARUM SOLL ICH,KOENNEN?          <101>
30043 DATA"GLAUBST DU, DAS BEEINDRUCKT MIC
      H, DASS DU",KANNST?                   <193>
39990 DATA$                                 <121>
39997 REM                                   <104>
39998 REM VERLEGENHEITS-TEXTE               <245>
39999 REM                                   <106>
40001 DATA REDEST DU IMMER SO EINFAELTIGES
       ZEUGS?                               <025>
40002 DATA ERZAEHLE MIR WAS DU VON MIR DEN
      KST.                                  <239>
40003 DATA"GLAUBST DU AUCH, DASS ICH DIR E
      TWAS UEBERLEGEN BIN?"                 <246>
40004 DATA WARUM SPRICHST DU EIGENTLICH MI
      T EINEM COMPUTER?                     <218>
40005 DATA KANNST DU NICHT ETWAS GESCHEITE
      RS ERZAEHLEN?                         <217>
40006 DATA BIST DU AUCH SO INTELLIGENT WIE
       ICH?                                 <042>
40007 DATA"UEBRIGENS: WAS HAELST DU EIGENT
      LICH VON UNSERER KONVERSATION?"       <162>
40008 DATA"WENN ES DIR ZU BLOED WIRD, ZIEH
       MIR EINFACH DEN STECKER RAUS."       <040>
40009 DATA EIN GESPRAECH MIT DIR IST ZIEML
      ICH EINFAELTIG.                       <234>
40010 DATA"UNSERER KONVERSATION NACH ZU SC
      HLIESSEN, BIST DU NICHT GERADE DER H
      ELLSTE!"                              <080>
40990 DATA$                                 <101>
40998 REM                                   <085>
40999 REM VERLEGENH.TEXTE AUF ?             <201>
41000 REM                                   <087>
41001 DATA DEINE FRAGEREI GEHT MIR AUF DEN
      WECKER!                               <155>
41002 DATA DUMME FRAGE! WEISS DOCH JEDER!   <127>
41003 DATA KEINE AHNUNG! WAS ERWARTEST DU
      EIGENTLICH VON MIR?                   <215>
41004 DATA DAS WEISST DU SICHER BESSER ALS
       ICH. ERZAEHLE WEITER!                <068>
49999 DATA$                                 <185>
```

Listing »VIC« (Schluß)



Markt&Technik-Buchverlag

Depot-Händler

Tragen Sie Ihre Buchbestellung und die Anschrift des Depotbuchhändlers auf die Bestellkarte in diesem Heft ein. Bitte vergessen Sie den Absender nicht.

Buchhandlung Herder, Kurfürstendamm 69
1000 Berlin 15, Tel. (5030) 8835002,
BTX *921782#
Computare Fachbuchhandlung, Keithstraße 18
1000 Berlin 30, Tel. (030) 2139021
Thalia Buchhaus, Große Bleichen 19
2000 Hamburg 36, Tel. (040) 3005050
Boysen + Maasch, Hermannstraße 31
2000 Hamburg 1, Tel. (040) 30050515
Electro-Data, Wilhelm-Heidsiek-Straße 1
2190 Cuxhaven, Tel. (04721) 51288
Buchhandlung Muehlau, Holtenauer Straße 116
2300 Kiel, Tel. (0431) 85085
ECL, Norderstraße 94-96
2390 Flensburg, Tel. (0461) 28181
Buchhandlung Weiland, Königstraße 79
2400 Lübeck, Tel. (0451) 74006-09
Buchhandlung Storm, Langenstraße 10
2800 Bremen 1, Tel. (0421) 321523)
Buchhandlung Lohse-Eissing, Marktstraße 38
2940 Wilhelmshaven, Tel. (04421)41687
Buchhandlung Schmorl u. v. Seefeld, Bahnhofstr. 13
3000 Hannover 1, Tel. (0511) 327651
Buchhandlung Graff, Neue Straße 23
3300 Braunschweig, Tel. (0531) 49271
Stern Verlag, Friedrichstraße 24-26
4000 Düsseldorf, Tel. (0211) 373033
Buchhandlung Baedeker, Kettwiger Straße 33-35
4300 Essen 1, Tel. (0201) 221381
Regensberg'sche Buchhandlung, Alter Steinweg 1
4400 Münster, Tel. (0251) 40541-5
Buchhandlung Acker, Johannisstraße 51
4500 Osnabrück, Tel. (0541) 28488
Buchhandlung Lensing, Westenhellweg 86-88
4600 Dortmund, Tel. (0231) 16980
Buchhandlung Meier + Weber, Warburger Str. 98
4790 Paderborn, Tel. (05251) 63172
Buchhandlung Phönix GmbH, Oberntorwall 25
4800 Bielefeld 1, Tel. (0521) 69071
Buchhandlung Gonski, Neumarkt 24
5000 Köln 1, Tel. (0221) 210528
Mayer'sche Buchhandlung, Ursulinerstraße 17-19
5100 Aachen, Tel. (0241) 48142
Buchhandlung Behrendt, Am Hof 5a
5300 Bonn 1, Tel. (0228) 658021
Buchhandlung Cusanus, Schloßstraße 12
5400 Koblenz, Tel. (0261) 36239
Akad. Buchhandlung Interbook, Fleischstraße 61-65
5500 Trier, Tel. (0651) 43596
Buchhandlung W. Finke, Kipdorf 32
5600 Wuppertal 1, Tel. (0202) 454220
Buchhandlung Balogh, Sandstraße 1
5900 Siegen, Tel. (0271) 55298-9
Buchhandlung Naacher, Steinweg 3
6000 Frankfurt 1, Tel. (069) 298050
Buchhandlung Wellnitz, Lautenschlägerstraße 4
6100 Darmstadt, Tel. (06151) 76548
Buchhandlung Feller + Gecks, Friedrichstraße 31
6200 Wiesbaden, Tel. (06121) 304911
Ferber'sche UNI-Buchhandlung, Seltersweg 83
6300 Gießen, Tel. (0641) 12001
Sozialwissenschaftliche Fachbuchhandlung, Friedrichstr. 24
6400 Fulda, Tel. (0661) 75077
Gutenberg Buchhandlung, Große Bleiche 29
6500 Mainz, Tel. (06131) 37011
Buchhandlung Bock + Seip, Futterstraße 2
6600 Saarbrücken, Tel. (0681) 30677
Buchhandlung Wilhelm Hofmann, Bismarckstraße 98
6700 Ludwigshafen, Tel. (0621) 516001
Buchhandlung Loeffler, B 1,5
6800 Mannheim 1, Tel. (0621) 28912
Buchhandlung Stehn, Bahnhofstraße 13
7000 Stuttgart 50, Tel. (0711) 561476
Buchhandlung am Markt, Kramstr. 6
7100 Heilbronn, Tel. (07131) 68682
PCB Micro-Computer, Oskar-Kalbfell-Platz 8
7410 Reutlingen, Tel. (07121) 270443
UNI Buchhandlung Kellner + Moessner, Kaiserstr. 18
7500 Karlsruhe, Tel. (0721) 691436
Buchhandlung Roth, Hauptstr. 45
7600 Offenburg, Tel. (0781) 22097
Rombach Center, Bertholdstraße 10
7800 Freiburg, Tel. (0761) 49091
Fachbuchhandlung Hofmann, Hirschstraße 4
7900 Ulm, Tel. (0731) 60949
Schauties Elektronik, Bachstraße 52
7980 Ravensburg, Tel. (0751) 26138
Buchhandlung Hugendubel, Marienplatz
8000 München 2, Tel. (089) 2389-1
Universitätsbuchhandlung Lachner, Theresienstr. 43
8000 München 2, Tel. (089) 521340
Buchhandlung Schönhuber, Theresienstr. 6
8070 Ingolstadt, Tel. (0841) 33146/47
Computerstudio Gertrud Friedrich, Ludwigstraße 3
8220 Traunstein, Tel. (0861) 14767
Buchhandlung Pustet, Kl. Exerzierpl. 4
8390 Passau, Tel. (0851) 56945
Buchhandlung Pustet, Gesandtenstraße 6
8400 Regensburg, Tel. (0941)53061
Buchhandlung Dr. Büttner, Adlerstraße 10-12
8500 Nürnberg, Tel. (0911) 232318
STS Computer Vertrieb, Werner-Siemens-Straße 19
8580 Bayreuth, Tel. (0921) 62320
Burger Elektro, Leimitzer Straße 11-13
8670 Hof, Tel. (09281) 40075
Sortiments- u. Bahnhofsbuchh. J. Strykowski, Bahnhofpl. 4
8700 Würzburg, Tel. (0931) 54389
Buchhandlung Pustet, Grottenau 4
8900 Augsburg, Tel. (0821) 35437
Kemptener Fachsortiment, Salzstraße 30
8960 Kempten, Tel. (0831) 14413

Belgien:
Eicher Micro & Personal Computer, Hünningen 56-58
B-4780 St. Vith, Tel. (080) 227393

Luxemburg:
Librairie Promoculture, 14, rue Duchscher (Pl. de Paris)
L-1011 Luxembourg-Gare, Tel. 480691, Telex 3112

Inserentenverzeichnis

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael M. Pauly (py)
Stellv. Chefredakteur: Michael Scharfenberger (sc)
Redakteure: aa = Albert Absmeier, leitender Redakteur, ev = Volker Everts, gk = Georg Klinge, hm = Harald Meyer, rg = Christian Rogge
Redaktionsassistenz: Gerda Vogl (202)
Fotografie: Janos Feitser/Jens Jancke, Titelfoto: Jens Jancke

Layout: Leo Eder (Ltg.), Dagmar Berninger, Willi Gründl

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Alpenstrasse 14, CH-6300 Zug, Tel. 042-223155/56, Telex: 862329 mut ch
USA: M & T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. (415) 424-0600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Herstellung: Klaus Buck (180)

Anzeigenverkaufsleitung: Ralph Peter Rauchfuss (126)

Anzeigenleitung: Christian Schlottau (282)

Anzeigenverwaltung und Disposition: Michaela Hörl (171)

Anzeigenformate: 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297x210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Anzeigenpreise: Es gilt die Anzeigenpreisliste Nr. 2 vom 1. Januar 1985.
Anzeigengrundpreise: 1/1 Seite sw: DM 8500,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1400,-. Vierfarbzuschlag DM 3800,-. Plazierung innerhalb der redaktionellen Beiträge: Mindestgröße 1/2-Seite
Anzeigen im Computer-Markt: Die ermäßigten Preise im Computer-Markt gelten nur innerhalb des geschlossenen Anzeigenteils, der ohne redaktionelle Beiträge ist. 1/1-Seite sw: DM 6400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. **Anzeigen in der Fundgrube: Private Kleinanzeigen** mit maximal 5 Zeilen Text DM 5,- je Anzeige.
Gewerbliche Kleinanzeigen: DM 11,- je Zeile Text.
Auf alle Anzeigenpreise wird die gesetzliche MwSt. jeweils zugerechnet.

Vertriebsleitung, Werbung: Hans Hörl (114)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebsgesellschaft mbH, Hauptstätterstraße 96, 7000 Stuttgart 1, Telefon (0711) 6483-0

Erscheinungsweise: 64'er, Magazin für Computerfans, erscheint monatlich, Mitte des Vormonats.

Bezugsmöglichkeiten: Leser-Service: Telefon 089/4613-119. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement verlängert sich zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

Bezugspreise: Das Einzelheft kostet DM 6,50. Der Abonnementspreis beträgt im Inland DM 78,- pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,- für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z.B. USA) um DM 38,-, in Ländergruppe 2 (z.B. Hongkong) um DM 58,-, in Ländergruppe 3 (z.B. Australien) um DM 68,-.

Druck: E. Schwend GmbH, Schmollerstr. 31, 7170 Schwäbisch Hall

Verantwortlich: Für redaktionellen Teil: Michael M. Pauly. Für Anzeigen: Christian Schlottau.

Redaktions-Direktor: Michael Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2,
8013 Haar bei München, Telefon 089/4613-0, Telex 522052



Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Hi-Eddi mit MPS 801 und MPS 802

Bilder von Hi-Eddi, dem fantastischen Zeichen- und Malprogramm aus 64'er 1/85, konnten bisher nur mit einem Epson-kompatiblen Drucker ausgegeben werden. Viele Leser fragten deshalb nach einer Druckerroutine für andere Drucker. Und sie sollten nicht vergebens warten. Wir veröffentlichen Routinen für MPS 801 (und kompatible, zum Beispiel Seikosha) und MPS 802/1526.

Textomat Plus

Schon seit langem angekündigt wurde die erweiterte Version des Textverarbeitungsprogramms Textomat von Data Becker. Wenn das Programm hält, was die Werbung verspricht, dürften keine Wünsche mehr offenbleiben. Aber Textomat Plus ist auch mehr als zweieinhalbmal so teuer wie der alte Textomat. Damit steigen natürlich auch die Ansprüche. Die Frage lautet also: Was leistet Textomat Plus wirklich und wie gut läßt sich mit ihm arbeiten?

Macro-Basic: mal was ganz anderes

Was stört eigentlich immer bei Basic-Erweiterungen? Man hat nie alles, was gebraucht wird. Nicht so Macro-Basic. Macro-Basic ist kein einzelnes Programm, sondern besteht aus zirka 160 verschiedenen kleinen Modulen. Mit diesen Modulen können Sie sich für jedes Programm Ihre eigene persönlich zugeschnittene Erweiterung zusammenstellen und verzichten damit auf unnötigen Ballast.

Tolle Preise für über 15000 Mark

Machen Sie mit bei zwei großen Wettbewerben! Es locken zwei Reisen nach Berlin, ein Btx-Fernseher und jede Menge Diskettenlaufwerke 1541.

Außerdem ...

- neue Turbo-Floppy getestet
- IEEE-Interface zum Selberbauen
- Lösung von Amazon
- und wieder viele Tips und Tricks für VC 20, C 16 und C 64

Dokumentation von Programmen leicht gemacht

Sie erinnern sich noch? Im 64'er, Ausgabe 12/84 riefen wir auf zum Erstellen eines Programms, daß bei der Dokumentation eigener und anderer Programme hilfreich sein sollte. Das Programm des Siegers ist wirklich gelungen. Es findet alle Variablen, die man zusätzlich kommentieren kann, zeigt sämtliche Sprungadressen und noch vieles mehr.

Datenfernübertragung für jedermann

Die Datenfernübertragung (DFÜ) nimmt einen immer größeren Stellenwert im Leben eines modernen Menschen ein. Deshalb sagen wir Ihnen, wie Sie mit Datex-P, Btx, Telebox und ISDN arbeiten, wie teuer das wird, und ob es sich überhaupt lohnt.



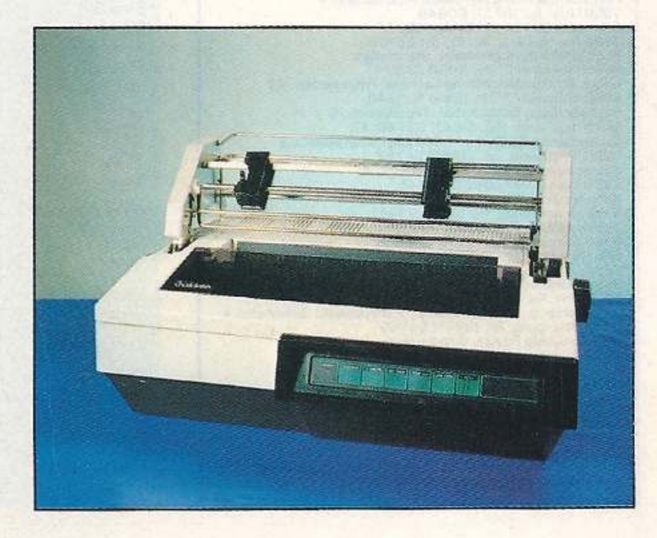
Drucker lernt das Lesen

Mit wenigen Handgriffen können Sie einen Epson-Drucker in den billigsten Scanner umrüsten, den es je gegeben hat. Mit dieser Hardwareerweiterung für den C 64 läßt sich alles, was auf ein DIN-A4-Blatt paßt, digitalisieren. Mit Hi-Eddi können Sie dann der Computer-Grafik den letzten Schliff geben.



Langsam, aber schön

Typenraddrucker sind eine Klasse für sich: Mit ihrem gestochen scharfen Schriftbild sind sie für jede Art von Korrespondenz geeignet. Besonders interessant sind Schreibmaschinen, die auch als Drucker verwendet werden können.




64ER ONLINE